MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH”

5-6

Bajtek

NR INDEKSU
PL ISSN 0860-1674

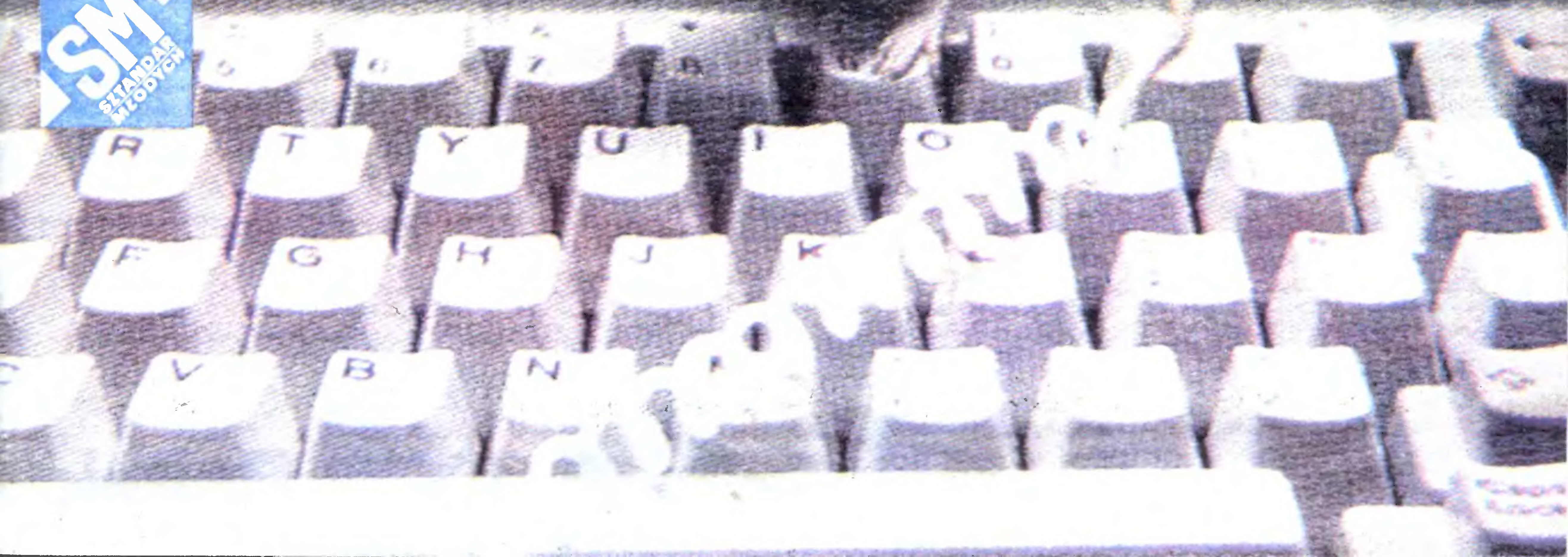
NR 3—4/86

CENA 100 ZŁ



FAIRLIGHT– gra roku!





A JEDNAK SZKOŁA!

Ciągle otrzymujemy listy, których autorzy proszą o pomoc w zdobyciu lub przynajmniej w uzyskaniu dostępu do komputera osobistego. „Jak mam przejść z mikrokomputerem na ty, do czego BAJTEK zachęca, skoro moich rodziców nigdy nie będzie stać na tak duży wydatek?" — żali się w liście do redakcji 13-letni Dariusz Grębkowski z Rzeszowa. „Kochani, wydrukowaliście artykuł, w którym Adam Krauze chwali się, że poznał tajemnice informatyki, gdyż zdobył IBM PC. Ale kogo stać na taki komputer!? Zastanówcie się co lansujecie! Przecież po takim artykule można się tylko skręcić z bezsilności!" — ma do nas pretensje Krzysztof z Warszawy.

Listy takie z jednej strony cieszą, gdyż świadczą przecież o autentycznym zainteresowaniu informatyką, ale również prowokują do refleksji mniej optymistycznych. Refleksji dotyczących społecznych konsekwencji „polskiej drogi" do informatyki popularnej.

Sytuacja jest otóż na dzisiaj taka, że — za sprawą komputerów osobistych — pogłębia się w błyskawicznym tempie rozwarstwienie społeczne wśród młodzieży. Mówiąc wprost: dzieci z rodzin bogatszych dostają coraz nowsze modele komputerów, a jednocześnie dzieci z rodzin biedniejszych mają do tego wymarzonego cuda techniki drugiej połowy XX wieku coraz dalej. Jak to ma się do haseł o stwarzaniu młodzieży równych szans startu, demokratyzacji systemu oświatowego itp. każdy czytelnik sam potrafi sobie odpowiedzieć.

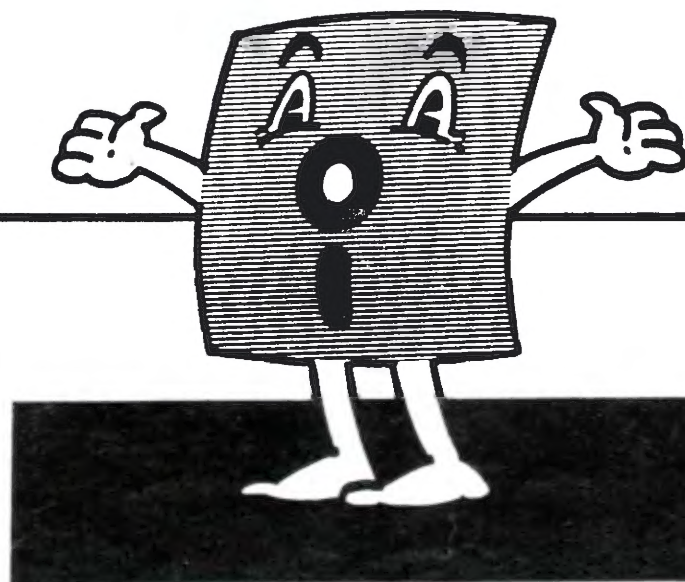
Może ktoś zapytać: Czy jest coś złego w tym, że ci rodzice, których na to stać, fundują swoim kilku i kilkunastoletnim pociechom „zabawki" o cenie dochodzącej, po przeliczeniach, nawet do miliona złotych? Nic złego, oczywiście, w tym nie ma, a nawet może to być z pewnością zakup o wiele bardziej sensowny niż kilka innych, które można byłoby za tę sumę poczynić. Problem polega na tym, że tych uprzywilejowanych jest niewielu — ok. 10 procent w skali kraju. Inni — czyli znakomita większość — takich „rodzinnych" szans na zdobycie komputerów nie mają i za swej młodości już raczej mieć nie będą. Dlatego konieczne są znaczące korekty na krętej drodze rozwoju naszej informatyki.

O powszechnej edukacji informatycznej będzie można mówić dopiero wówczas, gdy trafi ona szeroką falą tam, gdzie powinna być tzn. do szkół. Jak wiele nie mielibyśmy bowiem do naszego systemu oświatowego zastrzeżeń, jak wiele nie kierowalibyśmy pod adresem nauczycieli pretensji — faktem pozostaje, że tylko szkoła może zapewnić to, co w powszechnej edukacji jest najważniejsze, właśnie POWSZECHNOŚĆ. Tak właśnie, poprzez szkołę, naucza się informatyki zarówno na Wschodzie, jak i na Zachodzie. U nas również ograniczone nauczanie informatyki ma zostać wprowadzone od nowego roku szkolnego, ale jest to próba zbyt nieśmiała i pozbawiona niezbędnego rozmachu.

Mogę zapewne usłyszeć w tym momencie sto argumentów wykazujących, że szkoły i nauczyciele nie są do nauczania informatyki przygotowani, że brakuje komputerów i podręczników, że niski poziom nauczania może zniechęcić uczniów do informatyki i komputerów... Wszystko to być może prawdą. Ale nie wolno wyciągać z tego wniosku o poddaniu się, o zaniechaniu szerszych działań, tylko wniosek przeciwny: należy bowiem wykorzystać informatykę do zmiany oblicza szkoły. Jest to realne. Pasjonatów edukacji informatycznej w Polsce nie brakuje. Szkoła musi odważyć się, szeroko otworzyć drzwi i zaprosić ich do siebie. Oni już podpowiedzą skąd i na jakich warunkach zdobyć lub wypożyczyć sprzęt, którego będzie zresztą z roku na rok przybywać. To właśnie szkoła powinna mieć pierwszeństwo w zakupach i dostawach komputerów osobistych. Te same 10 maszyn przekazanych do małego klubu zaspokoi ciekawość 20-30 chętnych, gdy tymczasem oddane do klubu szkolnego stworzą szansę edukacji dla o wiele większej liczby osób. A co stoi na przeszkodzie, aby na przykład tym, którzy w edukacji komputerowej wykażą największe postępy i talent, szkoła podarowała w prezencie komputer osobisty?... Powiedzmy: jeden rocznie na klasę. Że za drogo? Po pierwsze — w ten lub inny sposób pieniądze na informatykę i tak będą z państwowej kiesy iść, a po drugie — to jest akurat inwestycja, która się stokrotnie w przyszłości zwróci (jak zresztą każda inwestycja w ludzi).

Jest szansa — dzięki edukacji informatycznej — przybliżenia szkoły do życia. Jest szansa wyrównania uczniom warunków startu życiowego, o których ciągle tak dużo się mówi. Są ludzie gotowi podjąć się tej pracy, gotowi wesprzeć szkołę... Jest nieźle wykształcona armia nauczycieli matematyki i fizyki, która — po odpowiednim przeszkoleniu — może również wziąć na siebie spory ciężar wysiłku w tej dziedzinie. Trzeba tylko jasno określić społeczne priorytety i drogowskazy. Jeśli ich nie postawimy, to o kształcie polskiej drogi do informatyki w dalszym ciągu decydować będą tylko prywatne portfele i konta.

Waldemar Siwiński



WYBIERZ SAM

ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu), Oskar Bramski, Krzysztof Czernek, Wiesław Migut, Sławomir Polak, Roman Poznański, Wanda Roszkowska (opr. graficzne), Marcin Waligórski, Roman Wojciechowski.

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH".
ADRES: 00-687 Warszawa, ul. Wspólna 61. Telefon 21-12-05.
Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych".
WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 100 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.
Zam. nr 714/86, nakład 200.000 egz. P-101



Bajtek



FOT. LEOPOLD DZIKOWSKI

OPTYMISTA

Rozmowa z Andrzejem Wojciechowskim, konstruktorem w Zakładach Urządzeń Komputerowych „MERA-ELZAB", zdobywcą I miejsca w Turnieju Młodych Mistrzów Techniki w woj. katowickim za opracowanie „Zmniejszenia wsadu pamięci EPROM 2716 do monitorów 7953 NCH/7209.

— **Co sądzisz o MERITUM?**

— Pracuję w zakładach, które go produkują i wolałbym się nie wypowiadać...

— **Nie chcesz narazić się dyrekcji lub kolegom?**

— Nie o to chodzi. Ja patrzę na mikrokomputer pod kątem profesjonalnym. Do pracy konstrukcyjnej potrzebny jest mi sprzęt o odpowiednich parametrach. MERITUM 1 tych wymagań nie spełnia, chociażby dlatego, że jest przeznaczony do innych celów. Natomiast może być z powodzeniem wykorzystany do celów edukacyjnych w szkole podstawowej, średniej czy w klubie.

— **MERITUM według założeń ma być komputerem domowym. Tylko... nie ma go w domach i przy obecnym tempie produkcji nie ma co liczyć, że to się zmieni. Jakie trzeba pokonać bariery, żebyśmy wreszcie mieli rodzimy, w miarę tani i powszechny, na średnim poziomie komputer personalny?**

— Może łatwiej byłoby wymienić, z czym nie mamy kłopotów. Uważam, że jest w kraju bardzo dużo fachowców na dobrym, światowym poziomie. Moi koledzy, podejmujący pracę za granicą, wcale nie ustępują wiedzą i fachowością specjalistom od hardware'u czy software'u. A jeszcze kilka lat temu w naszym zakładzie mało kto słyszał o mikrokomputerach. Dopiero ci, którzy kończyli politechnikę w roku 1979 lub 1980 mieli podstawy wiedzy teoretycznej. Co ważne, zainteresowania mikroinformatyką przestały być domeną ludzi z dyplomem uczelni. Dziś w moim zakładzie jednym z najlepszych specjalistów od uruchamiania MERITUM jest technik.

— Ale zainteresowania i wiedza to jeszcze mało. Przypomina mi się sprawa inż. Jacka Karpińskiego i jego komputera K-202. Był to pomysł nowatorski w skali europejskiej. Niestety, na początku lat siedemdziesiątych niemożliwy do zrealizowania przez nasz przemysł.

— Dlatego konieczne są decyzje o rozwijaniu produkcji części oraz doskonaleniu technologii. Oczywiście wiąże się to z odpowiednimi nakładami na elektronikę.

— Decyzje o rozwoju elektroniki podejmowano już kilkakrotnie w ciągu ostatnich kilkunastu lat.

— W ślad za decyzjami muszą iść konkretne działania. Tymczasem popatrzmy na produkowane u nas monitory lub mikrokomputery. Jakie elementy montowane są na płytkach? Polskich jest niewiele — i to najprostsze. Te o większej integracji głównie pochodzą albo ze Związku Radzieckiego albo z Zachodu. Do tego dochodzą nierzadkie kłopoty zaopatrzeniowe. Czasami, aby produkcja nie stała, wmontowuje się deficytowy element do płytki, następnie sprawdza, czy wszystko działa, potem wymontowuje, wsadza do następnej... i tak dalej.

— W tej sytuacji niedorzeczne albo prowokacyjne byłoby pytanie, czy jest szansa, abyśmy dogonili czołówkę.

— Lepiej zastanowić się, co zrobić, żebyśmy jeszcze bardziej nie zostawali w tyle. Niestety, jeśli nadal producenci sprzętu będą mieli do dyspozycji taką jak obecnie bazę elementową — to nie widzę żadnej szansy. Świat idzie do przodu — my stoimy. Trzeba rozwinąć produkcję elementów, a więc unowocześnić technologie wytwarzania elementów o dużej skali integracji: mikroprocesorów, elementów półprzewodnikowych. Musi być także udoskonalona technologia wytwarzania samych komputerów. Na przykład w naszym zakładzie tzw. wąskim gardłem jest klawiatura lub obudowa. Nie sprawia kłopotów druk czy opracowanie jego koncepcji!

— Jakie są bariery we wprowadzaniu do elektroniki nowoczesnych technologii?

— Zgodnie ze światowym trendem trzeba zmierzać do stosowania elementów umożliwiających większą ich integrację, co się z tym wiąże — większą niezawodność, mniejszą energochłonność itd. Słuszne jest także założenie, żeby stosować jak najwięcej krajowych elementów. Ale w tym miejscu zaczyna się dylemat: jakie robić projekty, gdy naszych elementów jest tak mało?

— Czy w takiej sytuacji informatyk lub konstruktor nie czuje się, tak jakby musiał po raz drugi wymyślać łopatę?

— Nie, jeśli inni mają dostęp do nowoczesnych elementów, nie mają ograniczeń technologicznych — to na pewno mają większe możliwości konstrukcyjne. I my korzystamy z tego, co inni już wcześniej wymyślili — nie błądzimy. Szkoda tylko, że tak mało jest obcej literatury i czasopism. Zachwycamy się katalogiem z 1982 lub 1983 roku, a wykorzystywany przez nas katalog o mikroprocesorze 8086 jest z 1976 roku...

— Taki dystans dzieli nas od przodujących firm?

— Zaczynamy produkować typ monitora, który masowo był robiony na Zachodzie w 1976 roku... Powiedzmy, dzieli nas dziesięć lat.

— Kiedy każdy będzie mógł postawić sobie na biurku mikrokomputer, chociażby takiej klasy jak MERITUM?

— Nie potrafię odpowiedzieć...

— Nawet w przybliżeniu?

— Mikrokomputery staną się powszechne wtedy, gdy będą tak tanie i tak proste w obsłudze jak długopisy. Lepiej zastanowić się, kiedy na ten sprzęt będzie mógł sobie pozwolić każdy, kto uzna, że jest mu potrzebny do nauki lub pracy. Ponieważ konstruktorów i programistów mamy niezłych — wszystko zależy więc od przemysłu. Byłbym szczęśliwy, gdyby nastąpiło to przed rokiem... dwutysięcznym.

— Czy to cię nie zniechęca?

— Nie, jestem wręcz optymistą! Komputer stanie się podstawowym narzędziem, którego człowiek będzie używał. Tak będzie na całym świecie — także u nas, chociaż z pewnym opóźnieniem. Dziś bardzo mylą się ci, którzy uważają, że elektronika to w gospodarce jeszcze piąte koło u wozu, można z nią poczekać, nie stać nas na jej rozwój, na inwestowanie. Nie można biec do tyłu, trzeba się odwrócić. Życie nas do tego zmusi.

— Jest to optymizm jakby nieracjonalny.

— Mam jeszcze teorię bardziej racjonalną. Otóż dziś ludzi można podzielić na trzy grupy. Tych, którym łatwo przychodzi znajomość z mikrokomputerem. Są to najczęściej młodzi, kończący studia, szkoły średnie. Do drugiej grupy należą starsi, którzy zasad działania mikrokomputera musieli uczyć się sami. Zdobywali nie tylko nową wiedzę, ale także przełamali opory psychiczne. Do trzeciej należą ci, którzy nigdy się tego nie nauczą — po prostu nie czują komputera. Zgodnie z prawami natury tych z grupy drugiej i trzeciej będzie coraz mniej.

Rozmawiał: Roman Wojciechowski

ELZAB·ELZAB·ELZAB

— Dlaczego to w ogóle robimy? — dyrektor przez chwilę szuka przekonującej odpowiedzi — Produkcja daje pracownikom nową wiedzę... Poza tym wykorzystujemy te urządzenia na różnych stanowiskach. I oczywiście mamy plany, zobowiązania, z których musimy się wywiązać...

Ale odpowiedź wydaje się mało przekonująca. Produkcja jest uboczna, jej wartość niewielka — zaledwie 1 proc. całej sprzedaży. A i sam produkt nie zbiera pochwał, zwłaszcza jeśli porówna się go z tym, co dziś powstaje na świecie. Jakby na pocieszenie zastępca dyrektora d.s. handlowo-technicznych **Zygmunt Korga** dodaje:

— **Wdrożenie było bardzo tanie, nie przekroczyło 10 mln złotych.**

Tyle właśnie kosztowało rozpoczęcie przez **Zakłady Urządzeń Komputerowych MERA-ELZAB** w Zabrzu produkcji pierwszego polskiego komputera personalnego **MERITUM I**. I rzeczywiście, 10 mln zł to kwota śmiesznie mała, jeśli uwzględni się cenę prac badawczych i wdrożeniowych w przemyśle elektronicznym. Prototyp pokazano publicznie jesienią 1983 roku na Targach Poznańskich. Fakt ten nie wywołał większego zainteresowania, ale cisza wokół MERITUM trwała krótko. Jeszcze przed rozpoczęciem seryjnej produkcji rozgorzała dyskusja nad „pierwszą jaskółką polskiej mikroinformatyki" — jak wówczas pisano. Tematem numer jeden były nadzieje na rodzimy, w miarę tani i dostępny komputer osobisty.

Minęły trzy lata i co pozostało z tych nadziei? Zanim jednak przejdziemy do dalszych losów MERITUM — parę słów o profilu produkcji MERY-ELZAB. Jedną gałąź stanowią urządzenia elektromechaniczne, jak **dziurkarka taśmy papierowej TT 105**, zestaw **dziurkarka-czytnik SPTP 3**. Były wprawdzie wątpliwości, czy na urządzenia tego typu będą jakiekolwiek zamówienia w latach osiemdziesiątych. Okazało się jednak, że ofert jest tyle, iż starczy ich na wypełnienie planów na całą pięciolatkę. Obecnie zakłady są jedynym producentem tych urządzeń w krajach socjalistycznych.

Drugi nurt produkcji — główny — to **monitory i terminale ekranowe w systemie monitorowym**. W ubiegłym roku MERĘ-ELZAB opuściło 15 tys. monitorów różnych typów. Mają one międzynarodowe badania i w większości są eksportowane. I w tym miejscu trzeba zatrzymać się na dłużej. Terminale, w pełni kompatybilne z urządzeniami produkowanymi na Zachodzie, od 1980 roku bazowały na procesorze **8080**. Takim typowo teletype'owym był monitor **7952** i późniejszy, pełny krajowy odpowiednik **DEC-owskiego VT 52** — w naszej nomenklaturze — **7953**. Obecnie wdrażany jest do produkcji model **79100**. Poszukiwania własnego develoupment systemu doprowadziły do powstania systemu **RTDS-8**, który został poszerzony o emulatory — **8085**, **Z80**.

Jak widać zakłady miały niezłe przygotowanie — zarówno pod względem kadry jak i bazy, aby przymierzać się do produkcji mikrokomputera. Dyrektor Korga, który przez siedem lat był konstruktorem w tej firmie, wspomina: *Gdy powstał RTDS-8, spodobał nam się pomysł, aby skonstruowac własny mikrokomputer. Zaczęlismy szukać pierwowzoru.*

Prace konstrukcyjne nad MERITUM rozpoczęły się w 1982 roku. Jako pierwowzór wykorzystano amerykański TRS-80, model II. Opracowany w 1978 roku, popularny w Stanach Zjednoczonych, w Europie był mało znany. W tym czasie coraz więcej było na światowym rynku ZX Spectrum, taniał Commodore. Dlaczego więc TRS-80? Tym bardziej, że od 1982 roku zaprzestano jego sprzedaży? Po prostu Basic TRS-80 był najbliższy RTDS-8. Był to model, który najłatwiej dawał się „przystosowac" do warunków naszej produkcji. Poza tym wykorzystano wcześniej przygotowaną klawiaturę do monitora teletype'owego.

Nie dość, że już w trakcie wdrażania do produkcji MERITUM było — określając delikatnie — niezbyt nowoczesnym urządzeniem — to jeszcze od roku

1984 produkuje się je w małych seriach. Dotychczas ok. 2,5 tys. sztuk! A świat uparcie ucieka.

Dlaczego tak mało? Najszybciej można to zrozumieć, podczas odwiedzin na **Wydziale Montażu Systemów Mikrokomputerowych**. Tu najlepiej widać, na czym polega uboczny charakter tej produkcji. Obok przestronnej hali, gdzie „wygrzewają się" **ComPAN-y**, niewielkie, ciasne pomieszczenie. **Kilka stanowisk montażowych, bardziej przypominających warsztaty rzemieślników niż stanowiska pracy w nowoczesnej firmie elektronicznej. Pracuje w tym pomieszczeniu brygada licząca... pięć osób: trzech „uruchamiaczy" i dwóch montażystów. Przy obecnej technologii, gdy otrzymują pakiet klawiatury i pakiet procesora, prace nad jednym egzemplarzem pochłaniają 20 roboczogodzin. Tegoroczny plan przewiduje produkcję ponad 2 tys. sztuk.**

— Ale jak tu myśleć o automatyzacji? — zastanawia się kierownik wydziału — **Andrzej Brodowicz** — Gdybyśmy robili dziennie 2 tys. MERITUM, to miałoby sens. Dziś nawet podwojenie produkcji wymagałoby zatrudnienia dodatkowo ok. 20 osób.

Jak rozwinąć produkcję? Dyrektor Korga odpowiada lapidarnie: — **Nie mamy żadnych możliwości, sprzedaliśmy licencję do... ELWRO**.

Druga sprawa — dość paradoksalna — to zamówienia. Po dwóch latach produkcji są wprawdzie zamówienia na dalsze dwa lata, ale w sumie opiewają na niezbyt imponującą liczbę... 4 tys. sztuk. Na wielkość zamówień wpływa m.in. to, że MERITUM — mimo iż produkowane jest na najlepszych, dostępnych częściach — jest modelem przestarzałym. Największym mankamentem są niedostatki grafiki. Zakłady pracy, którym potrzebne były komputery, zaopatrzyły się w urządzenia zagraniczne. Jedynym liczącym się odbiorcą mogłaby być oświata, ale MERITUM I nie nadaje się do celów dydaktycznych. W związku z tym w 1985 roku Zakłady złożyły Ministerstwu Oświaty i Wychowania propozycję: połączenie 8 MERITUM I w sieć, zarządzające byłoby MERITUM II, do tego monitor graficzny o dużej rozdzielczości. Oczywiście plus urządzenia peryferyjne. Dyskusja nad propozycją trwała rok. Wszystko rozbiło się o cenę.

Jakie będą więc losy pierwszego polskiego komputera personalnego? Dyrektor Korga twierdzi, że obecnie żaden zakład w kraju nie jest w stanie produkować więcej niż 5 tys. sztuk rocznie. Nie ma barier konstrukcyjnych: *„w Polsce można łatwo znaleźć kilka tysięcy inżynierów zdolnych opracować założenia nowoczesnego komputera"*. Największą barierą są części i wcale nie chodzi o „pamięci", a o zwykłe kondensatory, oporniki... Druga sprawa to cena. Wbrew tendencjom rynku światowego MERITUM nie tanieje, model 64kB bez stacji dysków i monitora kosztuje od roku 103 tys. zł. Na pocieszenie można dodać, że — tym razem wbrew tendencjom naszego rynku — nie drożeje. Ale zakłady do produkcji dopłacają, chociaż na razie niewiele, zależy z jakiego obszaru sprowadzają części. Dlatego najbardziej opłaci się produkować komputery profesjonalne, które można eksportować.

I jeszcze jedno: dziś nie jesteśmy w stanie produkować mikrokomputera, który miałby cenę konkurencyjną do giełdowej.

Jest jednak wiadomość na pocieszenie: MERA-ELZAB przygotowały nowy model: MERITUM III. Czym będzie różnił się od poprzednich? Tajemnica produkcji. Szefowie zakładów zainteresowanych zapraszają na Targi Poznańskie.

Roman Wojciechowski



FOT. LEOPOLD DZIKOWSKI



MERITUM II

Podstawowe dane techniczne:

Mikroprocesor U 880 D
— odpowiednik Z–80

Pamięć operacyjna	48 kB
Pamięć stała	8 kB
Pamięć obrazu	1 kB

— Organizacja obrazu — 64 znaki w 16 liniach lub 32 znaki w 16 liniach

— Reprogramowany generator znaków na PROM-ie

— Semigrafika — sygnały wyjściowe: zespolony sygnał wizyjny, sygnał wyjściowy w.cz. zmodulowany

Interfejsy — szeregowy w/g standardu RS 232C, trzy interfejsy równoległe we/wy z możliwością indywidualnego definiowania linii sterującej

Klawiatura kontaktronowa typu QWERTY

Komputer może pracować w czterech trybach pracy

— tryb bezpośredni (kalkulatorowy)

— tryb edycji

— tryb wykonywania programu

— tryb systemowy (do wprowadzania programów w języku maszynowym)

Mikrokomputer wyposażony jest w dwukieszeniową stację dysków produkcji NRD (K 5600) przystosowaną do dyskietek elastycznych o średnicy 5 1/4".

WOKÓŁ AMSTRAD-SCHNEIDER





Na zdjęciu: Joe Oki (z prawej) na stoisku Amstrada podczas międzynarodowej Wystawy Komputerów Biurowych, Domowych i Osobistych w Hotelu „Victoria" w Warszawie. Andrzej Łukomski (z lewej) dyrektor firmy Polanglia zajmującej się wysyłkową sprzedażą Amstradów do Polski.

ARNOLD

Alan Sugar jest Irańczykiem. Swą działalność gospodarczą rozpoczynał w Wielkiej Brytanii od montowania anten samochodowych. Dziś jego nazwisko zna cały komputerowy świat.

Joe Oki jest Japończykiem. Odebrał staranne wykształcenie. Cztery lata temu związał się ściśle z Alanem Sugarem. Ich dzieckiem jest ARNOLD.

ARNOLD — to imię jakie użytkownicy nadali w Wielkiej Brytanii komputerowi. Alan Sugar jest szefem potężnej firmy Amstrad Electronics, a Joe Oki — jego prawą ręką i dyrektorem do spraw eksportu. Kto jak kto, ale właśnie Japończycy najlepiej wiedzą jak sprzedawać.

Zaczynali od produkcji sprzętu hi-fi, który nie cieszył się zresztą zbyt wielkim poważaniem wśród nabywców. Interes rozkręcił się na dobre w chwili, gdy postanowili rozpocząć romans z komputerem. W ciągu dwóch lat wypuścili na rynek kilka wersji tego samego, ale z modelu na model coraz lepszego komputera — Amstrada. Najpierw CPC 464, potem na krótko 664, by przebojem wejść na rynek z modelem CPC 6128, pod koniec ubiegłego roku z PCW 8256 a w tej chwili — PCW 8512. W każdym z wymienionych modeli cyfry oznaczają pojemność pamięci RAM.

Kiedy 9 kwietnia br. miałem przyjemność poznać Joe Oki i zamienić z nim kilka słów, do Warszawy dotarła właśnie wiadomość, że „Amstrad Electronics" wykupił za 5 mln funtów (7,5 mln dol.) komputerową część przedsiębiorstwa „Sinclair Research" — producenta „Spectrum". Od tego też zacząłem naszą rozmowę.

— Serdecznie gratuluję interesu, jaki zrobiliście z Sir Clivem Sinclairem, ale równocześnie z troską i myślą o wszystkich polskich właścicielach „Spectrum" chciałbym spytać co będzie dalej z tym komputerem? Słyszałem, że ich produkcja ma być kontynuowana?

— Transakcja, o której Pan mówi, została zawarta w poniedziałek, 7 kwietnia. Tego samego dnia pan Sugar wrócił ze Stanów Zjednoczonych i widziałem się z nim tylko w przelocie. We wtorek odlatywałem do Warszawy — nie miałem więc okazji, by z nim porozmawiać, co dalej ze „Spectrum". Tak więc, po prostu nie wiem.

— Wizyta szefa Amstrada w Stanach Zjednoczonych związana była zapewne z zawartą wiosną br. transakcją na dostawę do USA 100 tys. komputerów Amstrad PCW 8256. Jaki jest cel pańskiej wizyty w Warszawie, czy jest to pierwsza wizyta w naszym kraju?

— Nie tylko pierwsza wizyta w Polsce, ale w ogóle w Europie Wschodniej. Przyjechałem, żeby obejrzeć zorganizowaną w hotelu „Victoria" wystawę komputerów, a głównie po to, by przekonać się, czy napływające do naszej firmy informacje o dużej popularności Amstradów w Polsce są zgodne z prawdą.

— I przekonał się Pan?

— To, co zobaczyłem na wystawie, nieustanny ścisk przy stoisku, na którym eksponowane były nasze komputery, w pełni potwierdził wcześniejsze opinie.

— O popularności jakiegokolwiek sprzętu komputerowego decydują nie tylko zalety hardware'u, lecz również dostępność programów. Czy są szanse na to, że wraz z przejęciem „Sinclair Research" przejmiecie również bogactwo oprogramowania do „Spectrum"?

— Firma „Amstrad Electronics" zajmuje się wyłącznie sprzętem. W komputerowym świecie, gdzie zmiany dokonują się w błyskawicznym tempie, nie wolno się rozpraszać. Sprawy softwaru pozostawiamy innym, i jak na razie — nie mamy żadnych podstaw do narzekania. Naprawdę martwimy się tylko o to, by utrzymać się w czołówce wyścigu do coraz lepszych, coraz większych i coraz tańszych komputerów.

*

„Amstrad Electronics" jest obecnie największym eksporterem sprzętu PC na rynku brytyjskim. Jako jedynemu chyba przedstawicielowi Europy Zachodniej udało mu się dostać na rynek amerykański z liczącą się dostawą sprzętu. Są na fali, a brytyjski odpowiednik naszego tygodnika „Veto" — „Which buy", w przeprowadzonej na początku br. ankiecie i oczywiście po wszechstronnych testach porównawczych uznał jeden z produktów firmy — CPC 6128 — godnym miana „The Best Buy" — najlepszy zakup. Hossę na sprzęt Amstrada potwierdza również stan finansów firmy. Doświadczenia innych każą jej jednak spoglądać na rynki wschodnie, jako potencjalnego odbiorcę ich komputerów. A może chodzi tylko o magazyny pełne „Spectrum", które komuś trzeba będzie sprzedać. Tylko komu?

Marek Car

W Wielkiej Brytanii ogłoszono niedawno o podjęciu prac nad piątą generacją komputerów przystosowanych do komunikacji głosowej z operatorem w kilku językach i symulujących procesy myślowe (**sztuczna inteligencja**). W pracach tych zaplanowanych na pięć lat uczestniczą: **ICL**, **Plessey**, **Uniwersytet w Manchester** i **Imperial College**.

• • •

Francuska firma **CTS** opracowała sześciokanałowy tuner TV umożliwiający wykorzystanie monitora komputerowego do odbioru programów telewizyjnych monochromatycznych i kolorowych. Tuner ten o nazwie **Synto X** przystosowany jest specjalnie do komputerów Amstrad 464, 664 i 6128.

• • •

Wielka Brytania i Francja podjęły współpracę w celu opracowania komputera do zastosowań naukowo-inżynierskich, zdolnego wykonywać 500 milionów operacji na sekundę i mającego kosztować 1/10 ceny współczesnych komputerów o podobnej mocy obliczeniowej. Prace projektowe mają potrwać trzy lata.

• • •

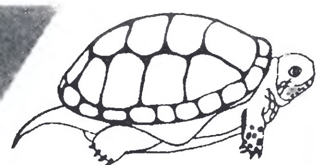
IBM przy współpracy kilku innych amerykańskich firm i uniwersytetów opracowała **" reduced-instruction-set-computer — RISC "**, czyli komputer o zredukowanej liście rozkazów. Mimo zwiększonej prędkości działania, uzyskanej dzięki uproszczeniu konstrukcji wewnętrznej. **RISC** jak narazie przeznaczony jest tylko do prac naukowo-inżynierskich (sztuczna inteligencja, grafika, operacje liczbowe).

• • •

Szwedzka firma **M Bejting AB** oferuje elektroniczny klucz zawierający program zapobiegający niepożądanym kopiowaniom programów. Zabezpieczenie o nazwie **ID-module** przyłączane do komputera jest specjalnym kodem, bez którego niemożliwe jest używanie jakichkolwiek programów.

• • •

Brytyjska firma **Miracle Technology Ltd** wprowadziła na rynek uniwersalny modem do komputerów Commodore 64 i 128. Modem umożliwia korzystanie z banków danych, poczty elektronicznej, telexu oraz komunikacji między użytkownikami. Cena na rynku brytyjskim ok. 100 funtów.



LOGO

LOGO — SŁOWNIK MINIMUM CZ. 2

Drugi odcinek wykazu procedur LOGO obejmuje definiowanie procedur oraz ich modyfikację, instrukcje wejścia/wyjścia oraz instrukcje warunkowe, iteracyjne i sterujące.

COPYDEF s s

Utworzenie nowej procedury o treści identycznej do treści jednej z uprzednio zdefiniowanych procedur LOGO. Pierwszy parametr jest nazwą tworzonej procedury, drugi — nazwą procedury kopiowanej.

```
TO ZMIEN.NAZWE :stara :nowa
COPYDEF :nowa :stara
ERASE :stara
END
ZMIEN.NAZWE "left "lewo
LEFT is a primitive in ZMIEN.NAZWE
```

DEFINE s l

Utworzenie nowej procedury o nazwie podanej jako pierwszy parametr i treści określonej przez parametr drugi — listę list, stanowiących kolejne wiersze definiowanej procedury. Jako pierwszy wiersz zawsze podaje się listę parametrów (w szczególności może ona być pusta).

```
DEFINE "czesc! [[][PRINT [Dzien
dobry, klaniam sie.]]]
PO "czesc!

TO CZESC!
PRINT [Dzien dobry, klaniam sie.]
END
```

DEFINEDP s → p

Funkcja określająca, czy dane słowo jest nazwą istniejącej procedury LOGO.

```
PR DEFINEDP "logo
FALSE
PR DEFINEDP "definedp
FALSE
```

EDIT sl — **ED sl**

Przejście do trybu redagowania procedur. Parametrem EDIT jest nazwa procedury*, którą chcemy redagować, lub lista takich nazw.

```
EDIT KWADRAT
ED [KWADRAT WIELOKAT OKRAG]
```

W wyniku wywołania ED ekran zmienia postać: jego dotychczasowa zawartość zostaje usunięta, na jej miejscu natomiast pojawia się tekst redagowanej procedury (procedur). Ostatni wiersz ekranu ma postać

```
LOGO EDITOR © SOLI / LCSI
```

W przypadku wywołania ED bez parametru na ekranie ukazuje się ostatnio redagowana procedura (lub „pusta strona", zależnie od implementacji LOGO na danym komputerze). Pusty ekran można również uzyskać przez wywołanie

```
ED []
```

W trybie redagowania tekst może być modyfikowany w dowolnym miejscu ekranu przy pomocy następujących klawiszy i ich kombinacji:

↑ ↓ ← →

Przesunięcie kursora o jedno miejsce we wskazanym kierunku;

DELETE (CAPS SHIFT — 0)

Usunięcie znaku na lewo od kursora lub pod kursorem;

EXTEND MODE ← EXTEND MODE →

Przesunięcie kursora na początek lub koniec wiersza;

EXTEND MODE ↑ EXTEND MODE ↓

Przesunięcie kursora na początek lub koniec strony (fragmentu tekstu o objętości jednego ekranu);

EXTEND MODE B EXTEND MODE E

Przesunięcie kursora na początek (B) lub koniec (E) redagowanego tekstu;

EXTEND MODE Y

Usunięcie części wiersza na prawo od kursora;

EXTEND MODE R

Wstawienie usuniętego przez ostatnio użyte **EXTEND MODE Y** tekstu w miejscu kursora;

EXTEND MODE P EXTEND MODE N

Wyświetlenie na ekranie poprzedniej (P) lub następnej (N) strony tekstu;

CTRL — L

Przesunięcie tekstu na ekranie tak, by kursor znajdował się w środkowej linii ekranu;

EXTEND MODE C — **CTRL — C**

Wyjście z trybu redagowania. Tekst wraz z dokonanymi poprawkami jest zapamiętywany jako treść procedur LOGO;

BREAK (CAPS SHIFT — BREAK) — **CTRL — G**

Przerwanie pracy w trybie redagowania. Wszystkie ostatnio wprowadzone zmiany są ignorowane. W zwykłym trybie Logo klawisz ten przerywa wykonanie jakiegokolwiek programu.

EDNS sl

Przejście do trybu redagowania nazw zmiennych oraz ich aktualnych wartości. W przypadku wywołania EDNS bez parametru redagowane są wartości i nazwy wszystkich aktualnie zawartych w pamięci zmiennych. Zob. EDIT.

```
MAKE "alfa [Snieg Deszcz Upal]
MAKE "beta 1.255
EDNS [alfa beta]
```

END

Znacznik końca definicji procedury. Zob. TO.

PRIMITIVEP s → p

Funkcja określająca, czy dane słowo jest nazwą procedury pierwotnej (primitive), czy nie.

```
PRINT PRIMITIVEP "primitivep
TRUE
PRINT PRIMITIVEP "logo
FALSE
```

TEXT s → l

Operacja, której argumentem jest nazwa istniejącej procedury LOGO, zaś wynikiem — jej treść w formie listy list, stanowiących kolejne wiersze procedury. Jako pierwszy wiersz zawsze występuje lista parametrów (w szczególności może ona być pusta).

```
TO HEJ
PRINT "Hej!!!
END
PRINT TEXT "hej
[][PRINT "Hej!!!]
```

TO s par1 par2.... — **TO s par1 par2....**

Początek definicji nowej procedury. Po słowie TO następuje nazwa tworzonej procedury oraz ewentualnie nazwy jej parametrów (każda poprzedzona dwukropkiem).

```
TO WIELOKAT :iloscbokow :dlugoscboku
```

Po wprowadzeniu tak konstruowanego nagłówka procedury znak zaproszenia z lewej strony nowego wiersza zmienia się z ? na >. Wpisywane teraz wiersze nie są wykonywane, lecz zapamiętywane jako tekst nowo tworzonej procedury. Treść definicji należy zakończyć wierszem, zawierającym wyłącznie słowo END.

```
TO WIELOKAT :iloscbokow :dlugoscboku
REPEAT :iloscbokow [FD :dlugoscboku
LT 360 / :iloscbokow]
END
WIELOKAT defined
```

Użycie klawisza **BREAK** w trybie definiowania procedury powoduje natychmiastowe wyjście z tego trybu i zignorowanie dotychczas wprowadzonych wierszy definicji.

2.2. Budowa algorytmu

IF p l — **IF pl l**
IF pl l → nsl

Instrukcja warunkowa. Pierwszym parametrem IF jest wyrażenie logiczne stanowiące warunek, drugim — lista instrukcji do wykonania w przypadku spełnienia warunku, ew. trzecim — lista instrukcji do wykonania w przypadku, gdy warunek nie jest spełniony.

```
IF MEMBERP "a [a b c][PR "Tak][PR "Nie]
Tak
```

Procedura IF może także pełnić role operacji.

```
PRINT IF MEMBERP "d [a b c]["Tak]["Nie]
Nie
```

W takim wypadku należy jednak użyć IF z trzema parametrami.

```
TO MAX :zbior
IF COUNT :zbior = 1 [OP FIRST :zbior]
OUTPUT IF (MAX BF :zbior) > (FIRST:zbior)
[MAX BF :zbior][FIRST :zbior]
END
PRINT MAX [10 0 4 25 1 -7]
25
```

IF p THEN l — **IF p THEN l ELSE l**
IF p THEN l ELSE l → nsl

IF w Terrapin Logo ma znaczenie analogiczne do IF z tym, że listy instrukcji są dodatkowo poprzedzone słowami THEN i (ew.) ELSE.

IFFALSE l — **IFF l**

Wykonanie podanej jako parametr listy instrukcji w przypadku, gdy warunek sprawdzony przez ostatnio użytą procedurę TEST przyjął wartość FALSE.

```
TO CZY :teza
TEST :teza
IFF [PR "Nie] IFT [PR "Tak]
END
CZY (2 + 4 > 5)
Tak
```

Wykonanie podanej jako parametr listy instrukcji w przypadku, gdy warunek sprawdzony przez ostatnio użytą procedurę TEST przyjął wartość TRUE.

```
TEST ("234 = 234) IFTRUE [PR "OK]
OK
```

OUTPUT nsl — **OP nsl**

Zakończenie wykonywania procedury i przekazanie jednostce nadrzędnej (innej procedurze) wartości, będącej parametrem OUTPUT. OUTPUT może występować wyłącznie w tekście procedury.

```
TO PI
OUTPUT 3.1415926
END
PRINT 2 * PI
6.2831852
TO L.MNOGA :slowo
OUTPUT WORD BUTLAST :slowo "i
END
PRINT L.MNOGA "noga
nogi
```



REPEAT n l
Powtórzenie listy instrukcji, stanowiącej drugi parametr ilość razy podaną jako parametr pierwszy.

```
TO GWIAZDA :iloscKatow :bok :st
REPEAT :iloscKatow [FD :bok
LT 360 * :st / :iloscKatow]
END
GWIAZDA 5 60 2
GWIAZDA 9 80 4
```

RUN l — **RUN l → nsl**
Wykonanie listy instrukcji podanej jako parametr. RUN może być również operacją.

```
TO LICZYMY
PR RUN READLIST
LICZYMY
END
LICZYMY
1 + SIN 90
2
```

STOP
Zakończenie wykonywania procedury. STOP może występować wyłacznie w tekście procedury.

```
TO ECHO :slowo
IF EMTYP :slowo [STOP]
PRINT :slowo
ECHO BUTFIRST :slowo
END
ECHO "Hej
Hej
ej
j
```

TEST p
Sprawdzenie wartości logicznej podanego jako parametr wyrażenia. Procedura używana w powiązaniu z IFTRUE lub IFFALSE.

```
TEST "slowo
SLOWO is not TRUE or FALSE
```

TOPLEVEL
Przerwanie realizacji programu.

```
TO POLE :a :b
IF OR (:a < 0)(:b < 0)[PR [Ujemny
bok prostokata?] TOPLEVEL]
OUTPUT :a * :b
END
PRINT POLE 10 5
50
PRINT POLE -10 10
Ujemny bok prostokata?
```

2.3. Wyjście

PRINT nsl — **(PR nsl ...)**
Wypisanie na ekranie monitora (lub, po uprzednim użyciu PRINTON, również na drukarce) podanego jako parametr obiektu. Przy wypisywaniu pomijany jest znak " poprzedzający słowo oraz zewnętrzne nawiasy list. Po zakończeniu wypisywania kursor jest przenoszony na początek nowej linii.

```
PRINT 345.8
345.8
PRINT SE "Numer 100
Numer 100
PRINT [ ALFA [Beta Gamma]]
ALFA [Beta Gamma]
```

PRINT1 nsl — **(PRINT1 nsl ...)**
Patrz TYPE.

SHOW nsl — **(SHOW nsl ...)**
Działa podobnie jak PRINT z tym, że listy drukowane są wraz z zewnętrznymi nawiasami, a słowa — poprzedzone znakiem ".

```
MAKE "alfa "beta
SHOW :alfa
"beta
SHOW SE :alfa 100
[beta 100]
PRINT SE :alfa 100
beta 100
```

SOUND l
Instrukcja pozwalająca na uzyskanie dźwięku w Sinclair LOGO. Parametrem SOUND jest lista dwuelementowa. Pierwszy element określa czas trwania dźwięku w sekundach, drugi — jego wysokość w półtonach (-62..75).

```
SOUND [10 24]
TO ZAGRAJ :slowo
IF EMPTYP :slowo [STOP]
SOUND SE 0.2 (ASCII FIRST :slowo) - 65
ZAGRAJ BUTFIRST :slowo
END
ZAGRAJ "Nabuchodonozor
```

TYPE nsl — **(TYPE nsl ...)**
Działa podobnie jak PRINT z tym, że po wypisaniu tekstu kursor nie jest przenoszony do nowej linii.

```
TYPE "Depeche PRINT "Mode
DepecheMode
```

2.4. Wejście

KEYP → p
Funkcja określająca, czy w danym momencie wciśnięty jest jeden z klawiszy.

```
TO JAZDA
FD 2
IF NOT KEYP [JAZDA]
IF RC = "p [RT 22.5]
IF RC = "q [LT 22.5]
JAZDA
END
JAZDA
```

RC? → p
Patrz KEYP.

READCHAR → s — **RC → s**
Operacja, wynikiem której jest słowo jednoelementowe, określające wciśnięty klawisz. READCHAR przerywa wykonanie programu do momentu, aż klawisz zostanie wciśnięty.

```
TO MASZYNA
TYPE READCHAR
MASZYNA
END
MASZYNA
```

READCHARACTER → s — **RC → s**
Patrz READCHAR.

READLIST → l — **RL → l**
Operacja, wynikiem której jest lista zawierająca tekst wprowadzony z klawiatury. Po wywołaniu READLIST na ekranie monitora pojawia się kursor, zaś wykonanie programu zostaje przerwane do momentu zakończenia wpisywania tekstu przez wciśnięcie ENTER (RETURN).

```
TO DIALOG
TYPE ":\  PR SE [Ja tez] RL
DIALOG
END
DIALOG
:lubie cukierki.
Ja tez lubie cukierki._
:
```

REQUEST → l — **RC → l**
Patrz READLIST.

2.5. Ekran tekstowy

CLEARTEXT — **CT**
Usunięcie wszystkich tekstów z ekranu. Fragment ekranu przeznaczony na grafikę pozostaje bez zmian.

```
TS REPEAT 10 [PR "Logo]
CLEARTEXT
```

CURSOR → l
Aktualne współrzędne kursora na ekranie w postaci listy dwuelementowej.

```
TS PR CURSOR
0 0
```

NODRAW
Patrz TEXTSCREEN.

SETCURSOR l — **SETCUR l**
Ustawienie kursora w żądanym miejscu ekranu. Parametrem jest dwuelementowa lista liczb, stanowiących numer kolumny (0..30) oraz wiersza (0..21).

```
TO PRZESUNKURSOR :dx :dy
SETCUR SE (:dx + FIRST CURSOR)
(:dy + LAST CURSOR)
END
TS REPEAT 6 [TYPE "X PRZESUNKURSOR 2 1]
```

TEXTSCREEN — **TS**
Przeznaczenie całej powierzchni ekranu na teksty; grafika zostaje z niego usunięta.

```
CS FD 40
TS PRINT "Halo
CS
TS
```

2.6. Atrybuty druku

BRIGHT n
Umożliwia wypisanie tekstu jaskrawym drukiem. Wywołanie BRIGHT 1 powoduje przejście do tego trybu; wyjście z niego następuje po wywołaniu BRIGHT 0, zakończeniu wykonywania programu lub pojawieniu się komunikatu o błędzie.

```
TO DOBITNIEJ :tekst
BRIGHT 1 TYPE (SE "\  :tekst "\ )
BRIGHT 0
END
TYPE [P.] DOBITNIEJ [mgr inz] PR
[Z. Nowak]
P. mgr inz Z. Nowak
```

FLASH
Przejście do trybu pisania na migającym tle; wyjście z tego trybu następuje wskutek wywołania NORMAL, zakończenia wykonania programu lub pojawienia się komunikatu o błędzie.

```
FLASH PRINT "Karol
```

INVERSE
Przejście do trybu pisania przy zamienionych kolorach tła i druku; wyjście z tego trybu następuje wskutek wywołania NORMAL, zakończenia wykonania programu lub pojawienia się komunikatu o błędzie.

```
INVERSE PR "Karol
```

NORMAL
Przywrócenie standardowych atrybutów druku, tzn. usunięcie efektów wcześniejszych wywołań FLASH i INVERSE.

```
TO W.KRATKE :slowo
IF EMPTYP :slowo [STOP]
INVERSE TYPE FIRST :slowo
IF EMPTYP BF :slowo [STOP]
NORMAL TYPE FIRST BF :slowo
W.KRATKE BF BF :slowo
END
W.KRATKE "Karol
```

OVER n
Umożliwia nakładanie wypisywanego tekstu na inny, uprzednio wypisany. OVER 1 powoduje przejście do trybu, w którym tekst na ekranie jest wypisywany bez usuwania tekstów już tam istniejących. Wyjście z tego trybu następuje po wywołaniu OVER 0, zakończeniu wykonywania programu lub pojawieniu się komunikatu o błędzie.

```
SETCUR [12 3] PR "Jablko
SETCUR [12 3] OVER 1 PR "Wisnia"
```

SETTC l
Ustalenie kolorów tła ekranu tekstowego i druku. Parametrem jest lista dwuelementowa liczb z zakresu 0..8, określająca te kolory.

```
SETTC [0 2]
PR [Czerwone i czarne]
```

---

*/ W różnych wersjach LOGO w różny sposób ustalono zasady pisowni słów, będących parametrami TO i EDIT; czasami należy poprzedzić je znakiem cudzysłowu ", kiedy indziej nie. W Sinclair LOGO prawidłowa jest forma
EDIT ALFA
TO ALFA
z kolei np. w DR LOGO na CPC 464 napiszemy
TO ALFA
ale
ED "ALFA

Marcin Waligórski

POKAŻ JĘZYK!

Może zabrzmi to trochę głupio, ale chciałbym ci dzisiaj pokazać język. Mam nadzieję, że nie czujesz się obrażony po przeczytaniu tej propozycji i pozwolisz mi się usprawiedliwić. Naprawdę nie miałem nic złego na myśli.

Jeśli chcesz prosić kolegę o jakąś przysługę musisz mu najpierw powiedzieć czego od niego oczekujesz. Do tego właśnie potrzebny jest **język**. Oczywiście nie musisz nic mówić, możesz na przykład napisać swoją prośbę na kartce lub wręcz pokazać gestem o co ci chodzi. Kolega będzie mógł ci pomóc jedynie wtedy, gdy zrozumie co chciałeś mu przekazać, to znaczy gdy będzie znał twój język. (Np. język gestów).

Ja — czyli komputer — jestem w takiej sytuacji jak twój kolega. Z tą tylko małą różnicą, że kolega — jeśli nawet zrozumie o co go prosisz — może ci odmówić, ja natomiast nigdy. Problem jest tylko jeden: wszystko musi być dla mnie zupełnie jasne. W gruncie rzeczy jestem strasznym głuptasem i nic sam nie wymyślę.

Moi pradziadkowie (komputery stają się dziadkami znacznie szybciej niż ludzie) programowani byli wyłącznie przez rozkazy skierowane do **procesora** — mózgu i serca każdej maszyny cyfrowej. Trzeba przyznać, że ten język nie miał wiele wspólnego z jakimkolwiek językiem porozumiewania się ludzi, a nawet w przypadku obliczeń matematycznych nie bardzo kojarzył się z tradycyjną arytmetyką. Nic więc dziwnego, że napisanie najprostszego programu wymagało sporej, specjalistycznej wiedzy.

Ale ludzie — zauważyłem — nie lubią się specjalnie przemęczać. Zaczęli więc tworzyć języki tzw. „**wyższego poziomu**", które coraz bardziej przypominają ich język naturalny. Niestety — jak mówią ludzie — nie ma róży bez kolców. Używanie tych języków przynosi obok wygody także niedogodności, z których największą jest znaczne zmniejszenie szybkości wykonywania programu. Dlatego też bezpośrednie programowanie procesora (**programowanie w języku wewnętrznym**) długo jeszcze będzie stosowane, szczególnie w przypadku bardziej złożonych programów takich jak skomplikowane programy graficzne.

Jak to się jednak dzieje, że potrafię zrozumieć polecenia wydawane mi w języku wyższego poziomu? Otóż posiadam (jak zresztą i większość moich kolegów) **interpreter języka BASIC**. Jest to coś w rodzaju „zaszytego" na stałe programu, który — podczas wykonywania programu napisanego w BASIC-u analizuje kolejno każdą linię i tłumaczy ją na rozkazy zrozumiałe dla mikroprocesora. Wymaga to jednak czasu. Znaczne lepszym rozwiązaniem są **kompilatory języków**. Ich działanie polega na tym, że cały program napisany w języku wyższego poziomu przekształca za pomocą kompilatora na język wewnętrzny i dopiero wówczas przystępuję do pracy. Jest to rozwiązanie o wiele szybsze.

Większość użytkowników domowych mikrokomputerów kojarzy sobie programowanie wyłącznie z językiem BASIC, czyli ciągiem rozkazów umieszczonych w numerowanych liniach, które wykonywane są przeze mnie w określonej kolejności. Nie jest to obraz prawdziwy. Wystarczy choćby popatrzeć jak wygląda programowanie w języku LOGO. Nie ma tu żadnej numeracji linii a jedynie programista uczy mnie kolejno wykonywania pewnych elementów (procedur), z których tworzy następnie bardziej złożone konstrukcje. O takim języku mówimy, że jest on strukturalny.

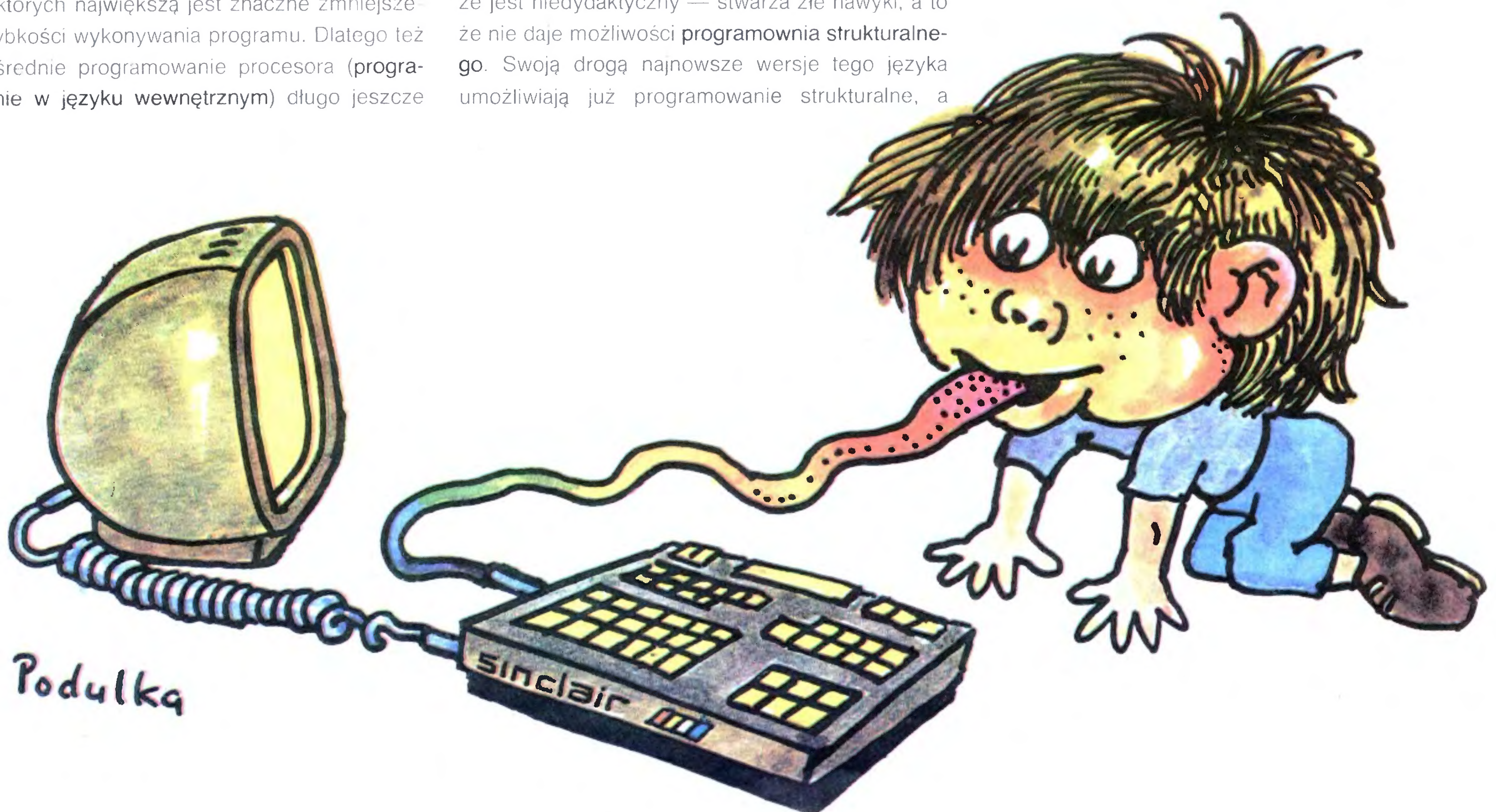
BASIC stał się moim (a także większości moich kolegów) podstawowym językiem właściwie przez przypadek. Stworzył standard, na który wybrzydzają prawie wszyscy profesjonalni informatycy. A to, że jest niedydaktyczny — stwarza złe nawyki, a to że nie daje możliwości **programownia strukturalnego**. Swoją drogą najnowsze wersje tego języka umożliwiają już programowanie strukturalne, a niektóre nawet nie mają numerowanych linii. Pozostaje jedynie pytanie: na ile jest to jeszcze stary, poczciwy BASIC.

Z drugiej strony, na rynku pojawia się coraz więcej **programów-kompilatorów** różnych języków. Dostępne na **dyskietkach**, **kasetach** i **kartridge'ach** (układach pamięci trwałej dopinanych do komputera) mogą być w każdej chwili wczytane do mojej pamięci i tym samym możesz się ze mną porozumiewać, w **PASCAL-u**, **PROLOG-u**, **LOGO** itp. Niektórzy z moich kolegów od razu potrafią porozumiewać się w dwóch językach. Wystarczy jedynie po uruchomieniu komputera wybrać ten, który nam odpowiada.

Pytasz po co komu tyle różnych języków? Czy nie można wybrać najlepszego a resztę wyrzucić na śmietnik? Niestety, nie można. Przyczyna zaś jest bardzo prosta: **NIE ISTNIEJE NAJLEPSZY JĘZYK**. Wszystko zależy do czego będę potrzebny. Innego języka będziesz używał przy obliczeniach inżynierskich. Innego zaś przy konstruowaniu baz danych czy badaniu wzajemnych relacji pomiędzy elementami zbioru. Języki przydatne w jednym przypadku mogą być bardzo niewygodne w użyciu (zarówno dla mnie jak i dla ciebie) w innym. Dlatego też warto poznać przynajmniej zasady programowania w kilku różnych językach.

Czego życzy ci z całego mikroprocesora

Twój Komputer



Amstrad — Schneider CPC 464

Pojemność pamięci RAM wynosi 64 kB, z czego 42 kB dostępne są dla użytkownika. Pamięć ROM — 32 kB zawiera BASIC oraz system operacyjny.

Amstrad 464 i jego bliźniak — Schneider produkowane są już od dwóch lat. Ich popularność — także w naszym kraju — jest bardzo duża. Amstrad sprzedawany jest głównie w Wielkiej Brytanii, Francji i Stanach Zjednoczonych, podczas gdy Schneider dominuje na rynku zachodnioniemieckim. Jedyna różnica między nimi polega na innym ustawieniu zwor elektrycznych na płytce, co powoduje pojawienie się różnych komunikatów po włączeniu komputera.

Mikroprocesor i pamięć

Amstrad — Schneider 464 pracuje na ośmiobitowym mikroprocesorze Z 80. Częstotliwość zegara wynosi 4 MHz. Z 80 jest najpopularniejszym mikroprocesorem stosowanym w komputerach domowych, co ma znaczenie dla wykorzystania istniejącej bazy programowej — CPC 464 ma możliwość implementacji programów w systemie CP/M 2.2.

Język

Językiem rezydującym w pamięci jest BASIC 1.0 firmy Locomotive Software. Został on opracowany na podstawie Microsoft BASIC dla IBM PC. Jest to z pewnością jeden z najbogatszych BASIC-ów w jakie wyposażone są mikrokomputery domowe. Zawiera praktycznie wszystkie (z niewielkimi wyjątkami) instrukcje komputerów Spectrum, Commodore 64, Atari 800XL i jeszcze kilka dodatkowych.

Oto kilka przykładów: AFTER i EVERY pozwalają na skok do podprogramu po upływie założonego czasu z dowolnego miejsca programu. SYMBOL pozwala bezpośrednio definiować znaki graficzne dostępne z klawiatury. RENUMBER przenumerowuje linie programu (lub jego części) z dowolnym skokiem i od dowolnej liczby. AUTO umożliwia automatyczną numerację linii również z dowolnym skokiem i od dowolnej liczby. Te dwie ostatnie komendy — stosowane powszechnie w profesjonalnych komputerach — oddają nieocenione usługi podczas samodzielnego pisania programów. BASIC 1.0 został wyposażony także w bardziej wyrafinowaną postać pętli — WHILE, WEND.

Również pod względem szybkości działania BASIC komputera CPC 464 stawia go w czołówce domowych komputerów.

Organizacja ekranu

Ekran monitora współpracującego z CPC 464 może być organizowany na trzy sposoby. W trybie normalnym mieści 40 kolumn i 25 wierszy tekstu, rodzielczość ekranu wynosi 320×200 punktów. W tym trybie można korzystać jednocześnie z 4 kolorów spośród 27, którymi dysponuje ten komputer.

W trybie wielokolorowym mamy jedynie 20 kolumn i 25 wierszy tekstu, 160×200 punktów, ale za to możliwość jednoczesnego stosowania 16 kolorów. Tryb wysokiej rozdzielczości pozwala natomiast na użycie tylko dwóch kolorów, lecz dysponujemy 80 kolumnami po 25 wierszy, 640×200 punktów.

Amstrad — Schneider posiada możliwość definiowania 8 okien tekstowych i jednego okna graficznego. Uzyskuje się je przy pomocy BASIC-owej instrukcji WINDOW.

Dźwięk

Dźwięk uzyskiwany jest przy pomocy 3 generatorów dźwięku i jednego generatora szumu pracujących na układzie AY-3-8912. Niestety, CPC 464 nie jest wyposażony w filtry obcinające zbędne pasma częstotliwości dźwięku. Mimo to można na nim pisać bardzo efektowne programy muzyczne, a nawet — z niezłym skutkiem — naśladować ludzką mowę.

Komputer wyposażony jest w głośnik zamontowany bezpośrednio w obudowie. Siła głosu regulowana jest za pomocą potencjometru. Podczas normalnej pracy czy zabawy z komputerem uzyskiwany dźwięk jest najzupełniej wystarczający, jednakże podczas realizacji programów muzycznych, wskazane jest podłączenie wzmacniacza do muzycznego wyjścia stereofonicznego (I/O).

Klawiatura

Klawiatura cyfrowa jest niezależna od cyfr na klawiaturze QWERTY. Może natomiast spełniać identyczną funkcję — jest to bardzo przydatne zwłaszcza przy wpisywaniu większej liczby linii danych (DATA) dla programów maszynowych. Może także służyć jako blok klawiszy funkcyjnych określonych przez programistę i wreszcie można przyporządkować jej dowolne dodatkowe znaki graficzne — np. polskie litery: ą, ę, ć, ż, ź..., pozostawiając bez zmiany cyfry na klawiaturze QWERTY.

W bloku klawiszy sterujących kursorem znajduje się klawisz COPY. Jest on bardzo przydatny pod-



FOT. LEOPOLD DZIKOWSKI

czas pisania i edycji — poprawiania programów. Pozwala on na przeniesienie do aktualnie pisanej lub edytowanej linii dowolnego fragmentu innej linii znajdującej się na ekranie. Operacja jest bardzo prosta i polega na sterowaniu dwoma niezależnymi



kursorami. Pierwszy — drukujący pozostaje w linii, którą piszemy, natomiast drugi — kopiujący kierujemy do linii, której fragment mamy zamiar skopiować.

Magnetofon

W tym modelu magnetofon stanowi integralną część komputera. Dla użytkownika, który zdecydował się na ten właśnie rodzaj pamięci masowej jest to rozwiązanie bardzo wygodne. Nie ma kłopotów z dodatkowym podłączaniem, zasilaniem, miejscem na stole. Z drugiej jednak strony, magnetofon — a właściwie jego układ mechaniczny — stanowi pod względem niezawodności najsłabsze ogniwo całego urządzenia. A o niezawodności systemu decyduje zawsze jego najsłabszy element.

Na szczęście jednak magnetofon, pomimo jego prostej konstrukcji (a może zwłaszcza z tego powodu) nie psuje się zbyt często. Przy zastosowaniu taśm przyzwoitej jakości nie ma też kłopotów z za-



pisywaniem i ładowaniem programów. Warto jedynie pamiętać, że komputer steruje magnetofonem włączając i wyłączając silnik. Nie jest natomiast cofana głowica. Gumowa rolka przez cały czas dociska taśmę do wałka napędowego. Nie należy więc zapominać o wyłączeniu magnetofonu za pomocą klawisza STOP natychmiast po zakończeniu nagrywania lub wczytywania. Warto także — przynajmniej najbardziej cenne programy — nagrywać tylko na jedną stronę kasety, pozostawiając drugą niezapisaną. Zagniecenie taśmy na końcu programu może bowiem uniemożliwić odczyt programu z drugiej strony kasety.

Zapis programu na kasetę może się odbywać z prędkością 1000 lub 2000 bodów (bitów na sekundę). Niestety, prędkość zapisu znacznie obniżają bardzo długie przerwy pomiędzy kolejno wczytywanymi blokami. Na plus należy zaliczyć natomiast możliwość podsłuchu poprzez wewnętrzny głośnik (regulacja głośności) zapisywanego czy ładowanego programu.

Magnetofon wyposażony jest również w licznik obrotów niezbędny przy katalogowaniu programów. Posiada także standardowe zabezpieczenie programów przed przypadkowym skasowaniem. Wygodny dostęp do regulacji głowicy umożliwia ustawianie jej podczas odtwarzania. Zdecydowanie odradzamy jednak jakiekolwiek samodzielne manipulacje!

Monitor i odbiornik TV

Amstrad — Schneider sprzedawany jest w zestawach z firmowym monitorem, w którym wbudowany jest również zasilacz do komputera (o napięciu 5 V). Kupującemu pozostaje jedynie decyzja czy wybrać monitor kolorowy — **CTM 640**, czy monochromatyczny z zielonym ekranem — **GT 64**. Ten drugi jest zresztą znacznie bardziej przydatny tym wszystkim, którzy mają zamiar korzystać z komputera nie tylko bawiąc się grami. Praca na tekstach (szczególnie w trybie wysokiej rozdzielczości) w przypadku stosowania kolorowego monitora męczy wzrok i jest niesłychanie uciążliwa.

Z drugiej strony gry komputerowe pozbawione



koloru stają się mniej atrakcyjne a niektóre całkowicie bezużyteczne. Jest na to sposób, pod warunkiem jednak, że posiadamy kolorowy **telewizor** pracujący w systemie PAL. Wystarczy wówczas dokupić modulator — **MP 1**, poprzez który podłączymy nasz komputer do gniazda antenowego odbiornika TV. Modulator posiada również wbudowany zasilacz.

Drukarki

CPC 464 współpracuje z drukarkami wyposażonymi w interface w standardzie **CENTRONICS**. Producent poleca firmową drukarkę **DMP-1**, lecz nic nie stoi na przeszkodzie, by zastosować inne drukarki, np. **GP 500 CPC** lub **Gemini-10X**. Ta ostatnia zresztą wydaje się być w naszych warunkach szczególnie przydatna, tak ze względu na stosunkowo niską cenę, wygodę obsługi, uniwersalność (papier z perforacją — regulowana szerokość traktora, papier bez perforacji z rolki, pojedyncze kartki) i wreszcie możliwość zastąpienia oryginalnej taśmy, taśmą stosowaną w zwykłych maszynach do pisania.



Stacje dyskietek

Jeśli przestanie nam wystarczać pamięć masowa w postaci magnetofonu kasetowego, możemy uzupełnić zestaw o jedną lub dwie stacje dyskietek. Są one łączone szeregowo: jako pierwsza, stacja **DDI-1**, jako druga **FD-1**. Napędy do tych stacji produkowane są przez firmę **Hitachi**.

Stacje pracują z dyskietkami 3". Posiadają **jednostronny odczyt**, na **40 ścieżkach** mieści się **180 kB** informacji (z tego 2 kB przeznaczone są na directory — obsługę wybierania programu). Posiadanie stacji dyskietek umożliwia nam korzystanie z systemu **CP/M**, wersja **2.2** firmy **Digital Research**. Przy zakupie stacji producent dołącza dwie dyskietki: **CP/M 2.2** oraz pracujący w tym systemie język **LOGO**, (prawdę mówiąc jest to chyba najgorsza z wersji tego języka).

Wyjście użytkownika

Stacje dysków podłączane są do wyjścia użytkownika — **user port**. Do tego samego wyjścia można jednocześnie podłączać także inne urządzenia i układy takie jak **dodatkowa pamięć**, **modem telefoniczny**, **cartridge** i **interface'y**. Odbywa się to na zasadzie nakładania jednej wtyczki na drugą.

Drążki sterowe

Amstrad — Schneider posiada tylko jedno wejście na **drążek** sterowy. Może jednak pracować z dwoma takimi manipulatorami podłączonymi szeregowo. Firmowy drążek **JY-1** posiada wyjście do podłączenia drugiego, tym razem już standardowego drążka.

Oprogramowanie

CPC 464 posiada bardzo bogate oprogramowanie zarówno użytkowe jak i przeznaczone do zabawy, w olbrzymiej większości dostępne na kasetach. Oczywiście programy pracujące w **CP/M 2.2** są z natury rzeczy dostępne tylko na dyskietkach.

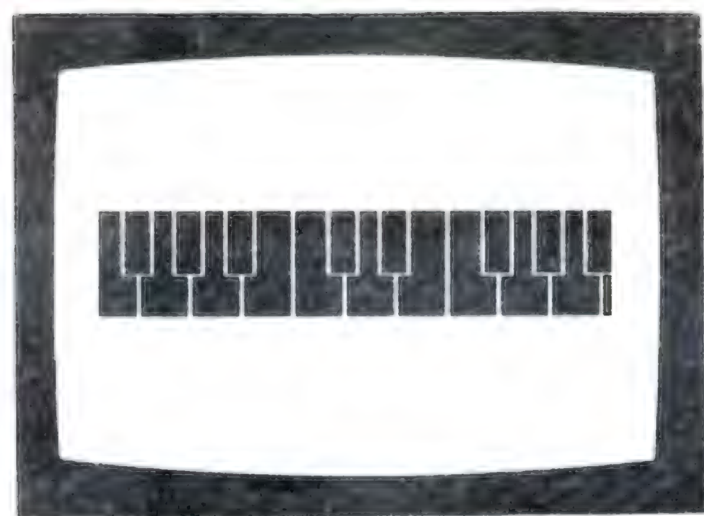
W ostatnich miesiącach także w Polsce pojawiło się sporo programów na ten komputer. Niektóre z nich — zwłaszcza użytkowe — zostały już przetłumaczone na język polski. Na przykład edytor tekstu — **TASWORD** baza danych DATEI czy program dydaktyczny „Zwierzęta, rośliny, minerały"

•

Podsumowując można stwierdzić, że porównanie Amstrada — Schneidera CPC 464 z innymi mikrokomputerami domowymi podobnej klasy wypada w większości przypadków na jego korzyść. Jednak bez wątpienia jest to jeszcze typowy komputer domowy, choć dysponuje pewnymi możliwościami stawiającymi go nieco wyżej.

Roman Poznański

MINI ORGANY



Może zainteresuje cię program, będący świetnym upominkiem dla młodszej siostry lub brata. Zapewnia on pełne wykorzystanie bogatych możliwości dźwiękowych Amstrada (Schneidera). Dwa najniższe rzędy klawiszy literowych zostają zamienione na klawiaturę instrumentu zgodnie z rysunkiem, który pojawia się po uruchomieniu programu. Pozostałe klawisze otrzymują następujące funkcje specjalne:

'1' — '9' — wybranie obwiedni tonu, czyli ustalenie, jak będzie zmieniać się wysokość dźwięku w czasie jego trwania
'f1' — 'f9' — wybranie obwiedni głośności: długość narastania, wybrzmiewania i opadania dźwięku
'↑' — zwiększenie szumu
'—' — zmniejszenie szumu
kursory:
'↑' — wyższa oktawa
'↓' — niższa oktawa
'→' — zwiększenie vibrato (wibracja dźwięku)
'←' — zmniejszenie vibrato

Nad rysunkiem klawiatury na ekranie pokazywane są stale aktualne parametry dźwięku: oktawa, vibrato, szum (O=brak szumu), numer obwiedni głośności i numer obwiedni tonu. Aby usłyszeć, jakie dźwięki może tworzyć Twój komputer, poeksperymentuj z tymi parametrami. Dla ambitniejszych podajemy niżej opis programu.

Opis programu:

zmienne:

okt— numer oktawy, jaką aktualnie reprezentuje klawiatura
vib— wartość vibrata, czyli głębokość modulacji tonu
obwg— numer obwiedni głośności
obwt— numer obwiedni tonu
szum— parametr określający rodzaj szumu: 0 = szum nie będzie wytwarzany
kor— zmienna służąca do obliczenia właściwej wysokości dźwięku w zależności od oktawy i parametru podstawowego
t— tablica parametrów podstawowych przypisanych poszczególnym klawiszom: na ich podstawie określa się wysokość dźwięku w danej oktawie
kan— numer kanału, do którego zostanie wysłany dźwięk + 128

program:

linie:

190–270— ustalenie kształtu dziewięciu obwiedni głośności; obwiednię aktualnie używaną do wytwarzania dźwięku określa zmienna obwg
310–340— ustalenie parametrów początkowych, wpisanie wartości podstawowych do tablicy t
380–400— nadanie kodów od 1 do 28 klawiszom dwóch dolnych rzędów klawiatury i klawiszom najwyższego rzędu
440–540— narysowanie klawiatury i wypisanie parametrów na ekranie
610–620— wybranie kolejnego kanału, dzięki tym dwom liniom kolejne dźwięki wysyłane są do kanałów A,B,C,A,B,C,A itd., co pozwala na jednoczesne zagranie trzech dźwięków. Znaczenie parametrów w linii 620: 999-wróć do kanału A; 129-wyślij dźwięk do kanału A, przedtem „wyrzuć" z niego poprzedni dźwięk, jeśli tamten jeszcze trwa (flush). 130, 132-analogicznie dla kanałów B i C
630— właściwa instrukcja wytworzenia dźwięku
650–720— zmiana parametrów: odpowiednim zmiennym zostaną przypisane nowe wartości i wykonany będzie skok do procedury wypisującej aktualne parametry: funkcje MIN i MAX zapobiegają przekroczeniu zakresu parametrów
800–840— ustalenie współrzędnych miejsca na ekranie, na którym będzie wypisana wartość wart
850— wypisanie wartości
900–980— podprogram ustalający kształt obwiedni tonu: obwiednie tonu muszą być od nowa podane komputerowi po każdej zmianie parametru vibrato
1030— procedura przywracająca standardowe znaczenie klawiszy i wypisująca komunikat o ewentualnym błędzie w programie.

Na podstawie „Sound mit dem CPC"
Schneider International 10/1985

Sergiusz Wolicki

```
10 '****************************
20 '*****                 ******
30 '*****   MINIORGANY    ******
40 '*****                 ******
50 '****************************
60 '
70 ' *** PROGRAM ZMIENIA ZNACZENIE NIEKT
ORYCH KLAWISZY. ***
80 ' *** UNIEMOZLIWIA TO NORMALNE UZYCIE
 KLAWIATURY W    ***
90 ' *** PRZYPADKU WYSTAPIENIA BLEDU LUB
 PRZERWANIA      ***
100 '*** PROGRAMU KLAWISZEM <ESC>.DLATEG
O PRZED ZATRZY- ***
110 '*** MANIEM PROGRAMU WYKONYWANA JEST
 LINIA 1030      ***
120 '
130 ON BREAK GOSUB 1030
140 ON ERROR GOTO 1030
150 '
160 '*** USTAWIENIE OBWIEDNI GLOSNOSCI.
 ***
170 '*** (NIE BEDA ZMIENIANE W PROGRAMIE
) ***
180 '
190 ENV 1,2,6,1,1,3,5,10,-1,10,5,-1,30,1
,0,200
200 ENV 2,1,15,1,15,-1,5
210 ENV 3,5,3,1,1,0,10,15,-1,3
220 ENV 4,1,15,1
230 ENV 5,1,15,1,15,-1,20
240 ENV 6,15,1,1,15,-1,20
250 ENV 7,1,15,1,1,0,20,1,-15,1
260 ENV 8,1,15,1,1,0,5,1,-15,1
270 ENV 9,1,15,1,15,-1,1,1,15,1,15,-1,1
280 '
290 '*** USTAWIENIE PARAMETROW POCZATKOW
YCH ***
300 '
310 vib=1:GOSUB 900
320 kor=0.0625:okt=1:obwt=1:obwg=1
330 DIM t(19):FOR n=1 TO 19:READ t(n):NE
XT
340 DATA 2863,2703,2551,2408,2273,2145,2
025,1911,1804,1703,1607,1517,1432,1351,1
276,1204,1136,1073,1012
350 '
360 '*** REDEFINICJA KLAWIATURY ***
370 '
380 FOR wart=1 TO 28:READ klawisz:KEY DE
F klawisz,0,wart:NEXT:SPEED KEY 10,1
390 DATA 71,60,63,61,62,53,55,54,44,46,4
5,38,39,36,31,29,30,28,22
400 DATA 64,65,57,56,49,48,41,40,33
410 '
420 '*** EKRAN ***
430 '
440 MODE 0:PEN 2:INK 1,26:PRINT "Oktawa
1  Vibrato 1",,,"ʃl Obwiednia gl←-os←'n.
 1ʃ0",,,"   Obwiednia tonu 1",,,"
      Szumy 0":PEN 1:LOCATE 5,11:PRINT "M
INIORGANY"
450 WINDOW 1,20,15,24:PAPER 1:PEN 4:CLS:
LOCATE 1,9:PRINT "z x c v b n m < > ?"
460 INK 4,16
470 FOR x=48 TO 640 STEP 64:MOVE x,0:DRA
WR 0,180,0:NEXT
480 FOR n=1 TO 10:READ a$:WINDOW 2*n,2*n
,15,20:IF a$="*" THEN 510
490 PAPER 7:CLS:LOCATE 1,5:PEN 4:PRINT a
$
500 MOVE 64*n-36,178:DRAWR 0,-100:DRAWR
36,0:DRAWR 0,100
510 NEXT
520 DATA s,d,f,*,h,j,*,l,":",";"
530 WINDOW 1,20,1,25
540 PAPER 0:PEN 1
550 '
560 '***** GLOWNA PETLA *****
570 '
580 ton$="":WHILE ton$="":ton$=INKEY$:WE
ND
590 v=ASC(ton$)
600 IF v>19 THEN 650
610 READ kan:IF kan=999 THEN RESTORE 620
:GOTO 610
620 DATA 129,130,132,999
630 SOUND kan,t(v)*kor,32767,0,obwg,obwt
,szum
640 GOTO 580
650 IF v=240 THEN kor=MAX(0.00390625,kor
/2):okt=MIN(5,okt+1):GOTO 800
660 IF v=241 THEN kor=MIN(1,kor*2):okt=M
AX(-3,okt-1):GOTO 800
670 IF v=242 THEN vib=MAX(0,vib-1):GOSUB
 900:GOTO 810
680 IF v=243 THEN vib=MIN(15,vib+1):GOSU
B 900:GOTO 810
690 IF v=45 THEN szum=MAX(0,szum-1):GOTO
 820
700 IF v=94 THEN szum=MIN(15,szum+1):GOT
O 820
710 IF v<58 AND v>48 THEN obwg=v-48:GOTO
 830
720 IF v>19 AND v<29 THEN obwt=v-19:GOTO
 840
730 GOTO 580
740 '
750 '*** KONIEC GLOWNEJ PETLI ***
760 '
770 '
780 '*** WYPISANIE ZMIENIONYCH PARAMETRO
W ***
790 '
800 x=7:y=1:wart=okt:GOTO 850
810 x=18:y=1:wart=vib:GOTO 850
820 x=18:y=7:wart=szum:GOTO 850
830 x=18:y=3:wart=obwg:GOTO 850
840 x=18:y=5:wart=obwt:GOTO 850
850 LOCATE x,y:PAPER 3:PRINT USING"##";w
art;:PAPER 0:GOTO 580
860 '
870 '*** USTAWIENIE OBWIEDNI TONU ***
880 '*** W ZALEZNOSCI OD VIBRATO  ***
890 '
900 ENT -1,1,vib,4,1,-vib,4
910 ENT -2,1,vib,2
920 ENT -3,1,-vib,2
930 ENT 4,1,5*vib,1,5,-vib,1
940 ENT 5,1,-5*vib,1,5,vib,1
950 ENT -6,1,vib,2,1,-vib,1
960 ENT -7,1,vib,9,1,-vib,9
970 ENT -8,1,vib,1,vib,-1,1
980 RETURN
990 '
1000 '*** BREAK/ERROR - inicjalizacja kl
awiatury i ekranu ; ***
1010 '***               wypisanie komuni
katu o bledzie      ***
1020 '
1030 CALL &BC02:CALL &BB00:PEN 1:PAPER 0
:MODE 2:PRINT "ʃlBla←,dʃ0";ERR;"(w linii
";ERL;")":PRINT:PRINT
Ready
```

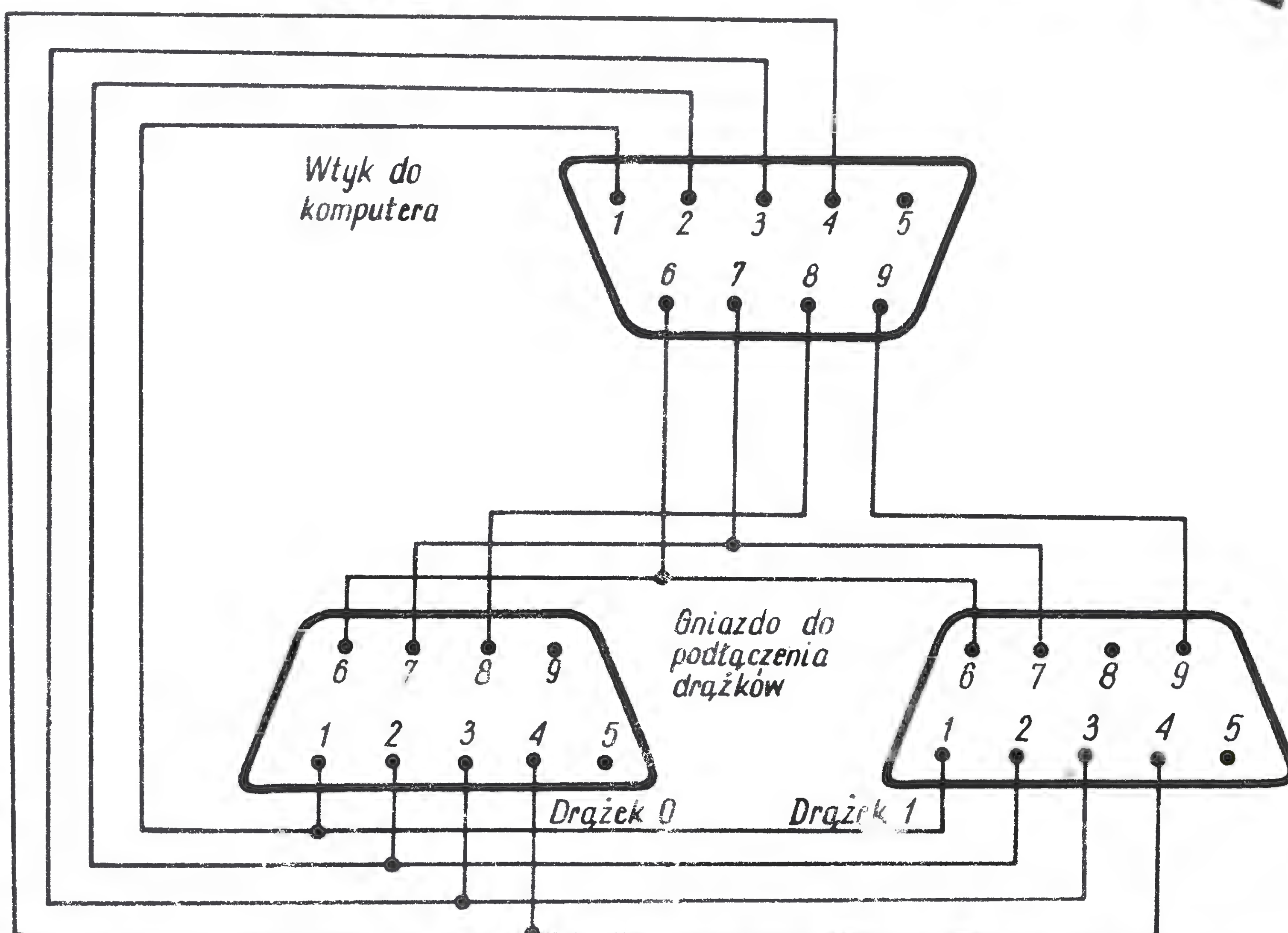
DRUGI DRĄŻEK STEROWY DO CPC 464 664 6128

Mikrokomputery z rodziny Amstrad-Schneider wyposażone są w jedno gniazdo do drążka sterowego. Nie znaczy to jednak, że komputer ten nie może pracować z dwoma drążkami. Sposób ich podłączenia natomiast jest inny niż np. w przypadku Commodore 64.

Oryginalne drążki Amstrada łączone są szeregowo — pierwszy posiada gniazdo do podłączenia drugiego. Co jednak zrobić, gdy posiadamy drążki bez takiego gniazda? Proponujemy wykonanie prostego urządzenia.

Jedyne co będzie nam potrzebne to wtyk do komputera (taki jak przy drążku sterowym) i dwa gniazda (takie jak w komputerze). Schemat połączeń elektrycznych przedstawiono na rys. 1. Gniazda najwygodniej jest zamontować w niewielkim pudełku z tworzywa sztucznego. Wtyk należy wyprowadzić na elastycznym, ośmiożyłowym przewodzie.

Roman Poznański



Rys.1. Schemat podłączenia dwóch drążków sterowych do mikrokomputera Amstrad-Schneider (widok gniazd i wtyczki od strony połączeń lutowniczych).

PISAĆ ZAMIAST PISAC

Oto krótki program umożliwiający posługiwanie się polskim alfabetem w trakcie pisania własnych programów na komputerach Amstrad 464 (664) 6128. Początkowo należy przepisać tylko linie od 11 do 14 i uruchomić ten fragment programu. Następnie wpisywać dalsze linie, zastępując duże litery w części opisowej (teksty w cudzysłowach) odpowiednimi literami już przy użyciu klawiszy funkcyjnych (6128) lub dodatkowej klawiatury cyfrowej (464/664)

f7 = ą	f8 = ę	f9 = ć
f4 = ł	f5 = ń	f6 = ó
f1 = ś	f2 = ź	f3 = ż

Po wpisaniu całego programu sprawdzić jego działanie i nagrać na dyskietkę. Linie numerowane są celowo od linii 11 i ze skokiem co 1, aby program nie kolidował z numeracją linii programu, w którym chcemy go zastosować (zasada numerowania ze skokiem co 10). Ułatwia to również aktualizację posiadanych już programów przez dopisanie polskich liter (kolejność: wgranie i uruchomienie naszego programu, następnie przy użyciu komendy CHAIN MERGE wgranie własnego, edytowanie linii wymagających zamiany liter i ich poprawienie przy użyciu klawiszy funkcyjnych).

Przyporządkowanie klawiszy funkcyjnych można zlikwidować wprowadzając sekwencję rozkazów w trybie bezpośrednim (linia bez numeru): **for i=1 to 9:key i, chr$ 48+i :next**. W programach wymagających wprowadzania wielu liczb wygodniej jest korzystać z klawiatury cyfrowej, która niestety blokowana jest przez ten program. Można jednak tego uniknąć, wykorzystując (z odpowiednio zmienioną numeracją) linie od 11 do 14 jako jeden podprogram, a linię kasującą przyporządkowanie (po wprowadzeniu do niej numeru) jako drugi podprogram, wywoływane z programu tworzonego przez użytkownika. A więc — do klawiatury!

Wojciech Ziółek

```
11 SYMBOL AFTER 129:
SYMBOL 137,16,8,60,1
02,96,102,60:SYMBOL
135,0,0,120,12,124,2
04,118,6:SYMBOL 136,
0,0,60,102,126,96,60
,12:SYMBOL 132,56,24
,28,24,56,24,60:SYMB
OL 133,4,8,216,102,1
02,102,102
12 SYMBOL 134,4,8,60
,102,102,102,60:SYMB
OL 130,4,8,126,76,24
,50,126:SYMBOL 131,8
,0,126,76,24,50,126:
SYMBOL 129,16,8,60,9
6,60,6,124
13 KEY 7,CHR$(135):K
EY 8,CHR$(136):KEY 9
,CHR$(137):KEY 4,CHR
$(132):KEY 5,CHR$(13
3)
14 KEY 6,CHR$(134):K
EY 1,CHR$(129):KEY 2
,CHR$(130):KEY 3,CHR
$(131):MODE 1:CLS
15 WINDOW #1,1,40,1,
3:PRINT #1," klawis
ze: f7=A";"     f8=E"
;"     f9=C":PRINT #1
,"                f4
=L";"     f5=N";"
f6=O":PRINT #1,"
          f1=S";"    f
2=Z";"     f3=Z"
16 WINDOW #3,1,40,4,
6:PRINT #3:PRINT #3,
"  Po skoNczeniu pis
ania programu usuN"
," linie od 15 do 17
 komendA DELETE 15-1
7":WINDOW #2,1,40,8,
25:PRINT #2:WINDOW S
WAP 0,2:PLOT 639,399
:DRAWR 0,-48:DRAWR -
639,0:DRAWR 0,48:
17 PLOT 8,344:DRAWR
627,0:DRAWR 0,-44:DR
AWR -627,0:DRAWR 0,4
4
```

ANIMACJA

Oglądając programy telewizyjne „Spectrum" i „Sonda" często podziwiamy, jak przed oczami projektanta obraca się na komputerowym ekranie wspaniały, trójwymiarowy przekrój samochodu. Napewno niejeden programista-amator marzy o stworzeniu takiego obrazu za pomocą mikrokomputera. Niestety, kiedy uruchamia on napisany mozolnie w BASIC-u program rzednie mu mina.

Wrażenie ruchu jakiegokolwiek większego przedmiotu jest bowiem znacznie osłabione przez migotanie całego rysunku. Wynika to ze sposobu, w jaki konstruuje się ruchome obrazy na ekranie. Najpierw trzeba narysować przedmiot w jednym położeniu, zmazać go, narysować w następnym, znowu zmazać itd. ... A wszystko to dzieje się przecież stosunkowo wolno.

Lepszym rozwiązaniem może być oczywiście zastosowanie języka maszynowego, lecz to dla odmiany wymaga sporej praktyki i znajomości komputera.

A gdyby tak można było mieć dwa obrazy umieszczone w dwóch obszarach pamięci i pokazywać je na zmianę. Moglibyśmy wtedy pokazywać jedno położenie przedmiotu z pierwszego obszaru, rysować następne w drugim, potem zaś zamieniać obszary rolami, rysując kolejne położenie w pierwszym, a pokazując drugi. Uniknęlibyśmy w ten sposób migotania obrazu i wrażenie ruchu poprawiło by się znacznie. Pomysł ten jest niestety nie do zrealizowania na Spectrum i wielu innych mikrokomputerach, gdyż tam ustalono położenie obszaru zarezerwowanego dla ekranu raz na zawsze i nie można go zmienić.

Realizacja idei, „dwóch ekranów" jest za to możliwa na komputerach Schneider (Amstrad). Tam adres pierwszej komórki, od której zaczyna się pamięć obrazu, przekazuje się specjalnemu układowi scalonemu CRTC (Cathode Ray Tube Controller — kontroler kineskopu). Adresuje on kolejne bajty pamięci pobierane następnie przez chip Gate Array (odpowiednik ULA ze Spectrum) i przetwarzane na sygnał telewizyjny. Poprzez użycie tylko dwóch rozkazów OUT dostępnych w BASIC-u możemy więc dowolnie zmieniać standardowe położenie ekranu (adresy &COOO- &FFFF). Druga ważna zaleta Schneidera polega na tym, że uzależnia on wartości adresów, pod które wysyła punkt grafiki i tekstu, od wartości zmiennych systemowych SCR BASE i SCR OFFSET. Dzięki temu, przypisując odpowiednie liczby tym zmiennym informujemy komputer, że powinien on rysować nie na normalnym ekranie, lecz na innym wybranym przez nas. W ten sposób dwoma rozkazami OUT i jednym POKE możemy zrealizować nasz pomysł.

Przykładem możliwości, jakie daje nam szybka zmiana obrazów, niech będzie program DOUBLESCREEN. Rysuje on trójwymiarowy sześcian lub czworościan i obraca go wokół pionowej osi symetrii. Złudzenie ruchu jest tu bardzo dobre. Po przeanalizowaniu działania tego programu proponuję poeksperymentować z różnymi rysunkami, na przykład kwadratu lub koła.

OPIS PROGRAMU:

— zmienne:

- scrbase — ustala adres zmiennej systemowej SCR BASE: jej wartość zależy od modelu komputera: dla CPC 464 wynosi &B1CB, dla CPC 664 i CPC 6128 &B7C6 odpowiednie przypisanie należy wykonać w linii 90
- fx.fy,fz — wyznaczają położenie obserwatora (perspektywę)
- mx,my,mz — wyznaczają położenie środka obrotu bryły
- k — długość krawędzi
- lw.lo — liczba wierzchołków bryły i liczba kolejno pokazywanych położeń
- a(i).b(i).c(i) — tablice pomocnicze
- u(n,i),v(n,i) — tablice zawierające współrzędne na ekranie i-tego wierzchołka w n-tym położeniu
- flag — O jeśli rysujemy na drugim ekranie, a pokazujemy pierwszy, -1 w przeciwnym przypadku
- proc — ustala, ktora procedura będzie użyta do rysowania bryły: 1 dla sześcianu, 2 dla czworościanu

— program:

linie:

- 50 — rezerwacja pamięci dla drugiego ekranu (&4000- — &7FFF), ustawienie opcji: kąty w stopniach zmienne standardowo typu całkowitego
- 90 — ustalenie parametrów początkowych; tu należy nadać wartość zmiennej scrbase zależnie od posiadanego modelu komputera
- 120 — ustalenie parametrów specyficznych dla danej bryły
- 170—190 — czytanie współrzędnych początkowych dla sześcianu z linii 630, dla czworościanu z linii 640
- 200—230 — zamiana współrzędnych przestrzennych na współrzędne ekranu i zapamiętanie ich w tablicach u (dla x) i v (dla y)
- 270—350 — umieszczenie napisu na obu ekranach; należy zwrócić uwagę na spacje w obu tekstach, tak aby napisy te wypadły w tym samym miejscu i przy zmianie ekranów wydawało się, że to jeden napis z migającym słowem DOUBLESCREEN
- 390 — wybranie kolejnych położeń bryły
- 430 — zamiana ekranów
- 520—550 — rysowanie sześcianu
- 590—690 — rysowanie czworościanu
- 650—690 — ustalenie standardowych parametrów przy przerwaniu pracy programu; skok do tej procedury nastąpi w wyniku instrukcji ON BREAK GOSUB z linii 50

— uwagi:

⎍ — specjalny znak sterujący, należy nacisnąć jednocześnie klawisze V i CTRL; analogicznie ← to H i CTRL

— w linii 90 można zmieniać wartości zmiennych i obserwować efekt na ekranie

— nie należy zapomnieć o wykrzyknikach przy nazwach niektórych zmiennych w liniach 170—210; oznaczają one, że zmienne nie są standardowo typu całkowitego, lecz typu rzeczywistego

— OUT &BCFF, &C wybiera rejestr układu scalonego CRTC
OUT &BDFF, &10 wpisuje do tego rejestru wartość oznaczającą, że należy czytać ekran od adresu &4000 OUT &BDFF, &40 przywraca standardowy adres &COOO

— w programie nie używamy zmiennej systemowej SCR OFFSET; komputer potrzebuje jej jednak normalnie do wykonywania scrollingu

Na podstawie „Bewegte Grafik mit drei Befehlen" Happy Computer 10/85

Sergiusz Wolicki

```
10 '                DOUBLESCREEN
20 '
30 ' ****** animacja prostych bryl geome
trycznych ********
40 '
50 MEMORY &3FFF:ON BREAK GOSUB 650:DEFIN
T a-z:DEG:MODE 2
60 '
70 ' ustalenie parametrow poczatkowych
80 '
90 fx=250:fy=-1000:fz=500:mx=320:my=200:
mz=150:k=200:flag=-1:scrbase=&B1CB
100 CLS: LOCATE 7,10: PRINT "∏1 szes←'ci
an (1) czy czworos←'cian (2) ? ∏0"
110 k$=INKEY$:IF k$<"1" OR k$>"2" THEN 1
10
120 CLS: PRINT "Momencik":IF k$="1" THEN
 lw=8: lo=9: RESTORE 630: proc=1 ELSE lw
=4: lo=12: RESTORE 640: proc=2
130 DIM a(lw),b(lw),c(lw),u(lo,lw),v(lo,
lw)
140 '
150 ' obliczenie wspolrzednych

160 '
170 FOR i=1 TO lw:READ x!,y!,z!
180 a(i)=x!*k:b(i)=y!*k:c(i)=z!*k
190 NEXT
200 FOR n=1 TO lo:si!=SIN(n*10):co!=COS(
n*10):FOR i=1 TO lw
210 x=a(i)*co!-b(i)*si!+mx:y=a(i)*si!+b(
i)*co!+my:z=c(i)+mz
220 u(n,i)=(fy*x-y*fx)/(fy-y):v(n,i)=(fy
*z-y*fz)/(fy-y)
230 NEXT:NEXT
240 '
250 ' poczatek rysowania
260 '
270 WINDOW #7,1,80,3,25
280 CLS:PRINT "∏1Przykl←-ad uz←'ycia pro
cedury 'DOUBLESCREEN':∏0"
290 PRINT "∏1Program ten zostal←- napisa
ny cal←-kowicie w Basic-u!∏0"
300 '
310 ' to samo na drugim ekranie (&4000-&
7fff)
320 '

330 POKE scrbase,&40:CLS
340 PRINT "∏1Przykl←-ad uz←'ycia procedu
ry '                 ':∏0"
350 PRINT "∏1Program ten zostal←- napisa
ny cal←-kowicie w Basic-u!∏0"
360 '
370 ' animacja
380 '
390 FOR n=1 TO lo
400 '
410 ' zamiana ekranow
420 '
430 flag=NOT(flag):OUT &BCFF,&C:IF flag
THEN OUT &BDFF,&10:POKE scrbase,&C0 ELSE
 OUT &BDFF,&30:POKE scrbase,&40
440 '
450 ' rysowanie
460 '
470 CLS #7:ON proc GOSUB 490,560
480 NEXT:GOTO 390
490 '
500 ' szescian
510 '

520 FOR i=1 TO 4:FOR j=5 TO 8
530 IF i+j<>9 THEN MOVE u(n,i),v(n,i):DR
AW u(n,j),v(n,j)
540 NEXT:NEXT
550 RETURN
560 '
570 ' czworoscian
580 '
590 FOR i=1 TO 3:MOVE u(n,4),v(n,4):DRAW
 u(n,i),v(n,i):NEXT
600 FOR i=1 TO 2:MOVE u(n,i),v(n,i):DRAW
 u(n,i+1),v(n,i+1):NEXT
610 MOVE u(n,3),v(n,3):DRAW u(n,1),v(n,1
)
620 RETURN
630 DATA -0.5,0.5,0.5,0.5,0.5,-0.5,-0.5,
-0.5,-0.5,0.5,-0.5,0.5,-0.5,0.5,-0.5,0.5
,0.5,0.5,-0.5,-0.5,0.5,0.5,-0.5,-0.5
640 DATA 0,0.577,-0.272,-0.5,-0.289,-0.2
72,0.5,-0.289,-0.272,0,0,0.544




650 '
660 ' BREAK - ustawienie standardowych p
arametrow ekranu
670 '
680 OUT &BCFF,&C:OUT &BDFF,&30:POKE scrb
ase,&C0
690 END
Ready
```

'86 MUNDIAL



Jeżeli chcesz zachować w pamięci swojego komputera sylwetki Diego Maradony, Michela Platini i innych idoli boiska weź program MUNDIAL CARICATURE!

Rysunki, które zobaczysz na ekranie monitora przekonają Cię, jak precyzyjna może być grafika ZX SPECTRUM!

FAIRLIGHT

Kraina Fairlight była niegdyś szczęśliwym i wesołym miejscem. Przed 3000 lat krajem tym rządził zacny król Avars. Po jego śmierci całe królestwo pogrążyło się w głębokiej rozpaczy. Światło skryło się w mroku, dni stały się szare i ponure, szczęście i pokój kraju chyliły się ku upadkowi.

Z biegiem lat jedynie zamek króla Avarsa pozostał samotny pośrodku równiny Avarslund, niedostępny i otoczony tajemnicą. Legendy mówią, że promienie słoneczne uwięzione w zamku oświetlają jego komnaty. Krążą także podania o nieśmiertelnym Segerze, który zamieszkuje mury zamczyska, czekając stosownej chwili, by zwrócić światło mieszkańcom krainy. Nikt jednak nie wie, kiedy tak żarliwie oczekiwana przez wszystkich Fairlightczyków chwila nadejdzie. Wielu było śmiałków, którzy próbowali sforsować niedostępne mury ponurego zamczyska i uwolnić światło. Żaden z nich jednak nie wrócił.

Pewnego dnia Isvar (bohater, którym kierujesz), siedział na skraju lasu otaczającego zamek Avarsa, rozmyślając nad tą historią. Las ten powszechnie uchodzi za straszne i niezwykle niebezpieczne miejsce, ponieważ stanowi tereny łowieckie Kobiety-Potwora o imieniu Ogri. W przypływie rozpaczy Isvar postanowiła dostać się do zamku i przywrócić szczęście i radość swoim rodakom. Lekceważąc ostrzeżenia zagłębił się w leśne ostępy. Nie uszedł jednak zbyt daleko, gdy z gęstwiny wyłoniła się krwiożercza Ogri i siłą zawlokła go do swojej groty, aby go pożreć. Z opresji wybawił Isvara tajemniczy starzec w kapturze — wielki mag i czarownik. Wyprowadził go z groty Ogri wprost pod mury owianego legendami zamczyska i wskazał tajemnicze przejście prowadzące do zamku.

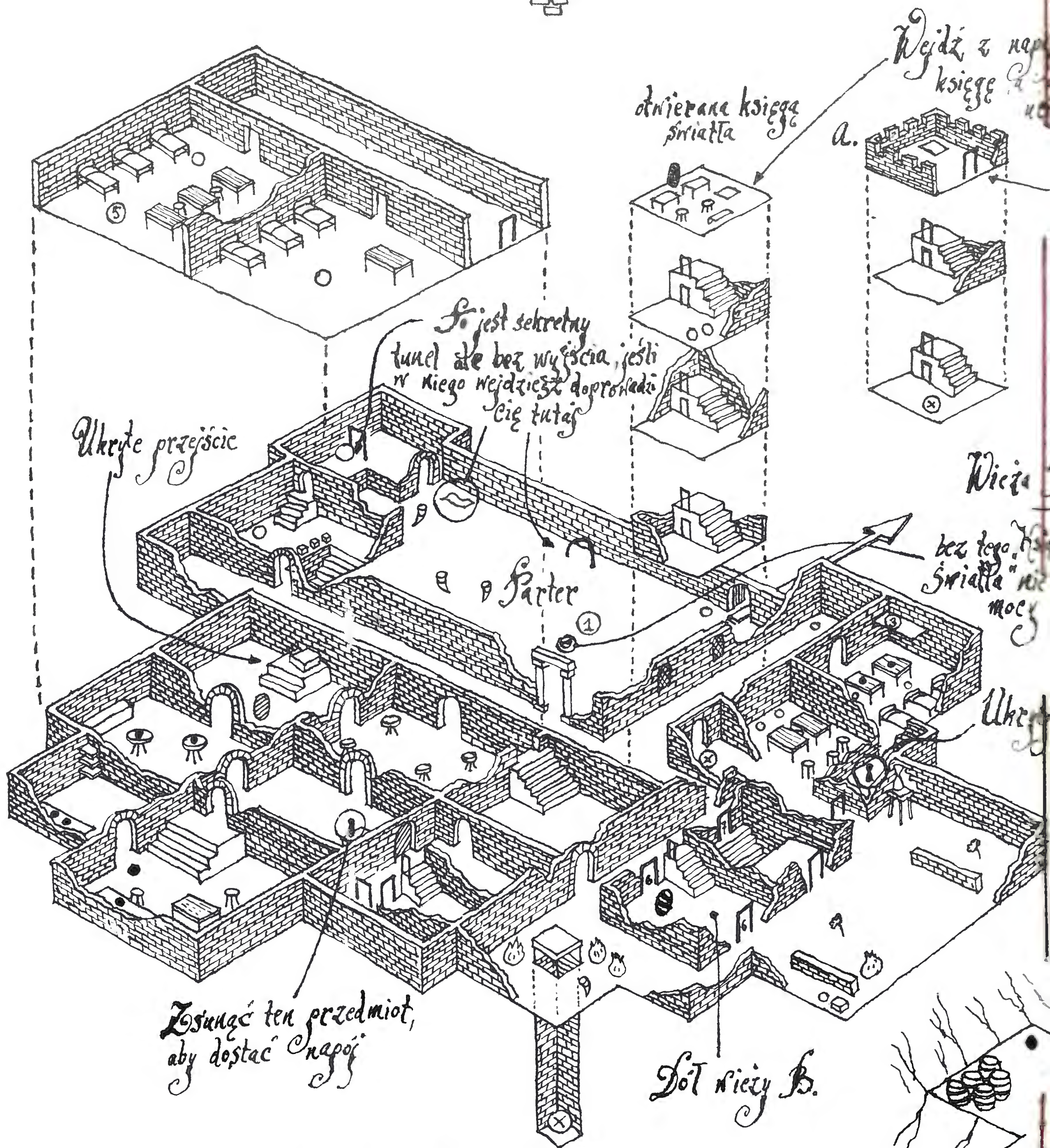
Starzec był królewskim czarodziejem uwięzionym w zamku Avarsa od tysięcy lat. Teraz również Isvar stał się królewskim więźniem. Starzec nim zniknął zdradził mu jednak pewną tajemnicę: wydostać się z zamku może jedynie ten, kto odnajdzie Księgę Światła ukrytą wśród murów i przyniesie ją czarodziejowi. W niej to bowiem zapisane zostało hasło uwalniające wielkiego maga. Aby jednak dojść do Księgi trzeba pokonać wiele niebezpieczeństw.

Isvar uzbrojony jest w szpadę. Może poruszać się w czterech kierunkach, skakać, podnosić znalezione przedmioty oraz walczyć z napotkanymi w drodze przeciwnikami. Może także przesuwać przedmioty oraz ustawiać jedne na drugich aby dostać się do wysoko położonych okien lub drzwi. Jego przeciwnikami są strażnicy, trolle, upiory, zjawy. Strzec się także musi wędrujących kielichów. Nawet jedna kropla z potrąconego kielicha zabije.

Zbierając poszczególne przedmioty należy pamiętać, że Isvar może ich nieść maksymalnie pięć. Każdy przedmiot ma jednak swoją wagę i może zdarzyć się, że nasz bohater udźwignie tylko dwa ciężkie przedmioty. Trzeba jednak wiedzieć, które z tych przedmiotów będą pomocne Isvarowi w spełnieniu jego misji. Będą to przede wszystkim klucze otwierające kolejne drzwi. Natomiast w walce z przeciwnikami przydadzą mu się na pewno magiczne krzyże, woreczki pełne złota i życiodajny napój.

Księga Światła ukryta jest na szczycie wieży, w której uwięziony jest królewski czarodziej. Leży schowana głęboko pod znajdującym się tam sarkofagiem. Aby dostać się do jego wnętrza niezbędna jest znaleziona wcześniej korona. Kiedy już uda się Isvarowi zdobyć księgę musi odnaleźć komnatę czarodzieja. Pamiętajmy także, aby do tego momentu zachować krzyże i napój, które będą bronią w razie ataku nieprzyjaciół. Gdy czarodziej ma już swój upragniony skarb Isvar może szczęśliwie opuścić niegościnne zamczysko (tym samym sekretnym przejściem, którym się tam dostał), uwalniając upragnione światło. Przywróci ono mieszkańcom Fairlight radość, pokój i szczęście.

Sławomir Polak

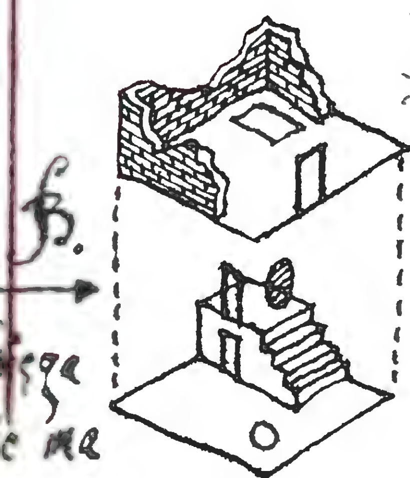


Fairlight – lan gry

Zygmunt Zaradkiewicz rys.

...yem lub krzyżem - weź klucz - daj magiczną
...n zamieni się w widmo - zabij go a potem
...ciekaj główną bramą

— przejście do wieży B.



...ęga
...e ma

... napój

legenda

- ○— ↦ przedmiot rytualny
- ↦ ukryte drzwi
- ● ↦ jedzenie
- ○ ↦ straż
- ↦ silny prąd powietrza
- ● ↦ olbrzym
- 8 ↦ klepsydra, która obezwładnia zwierzęta
- ⊗ ↦ bąbel
- ① ↦ klucz
- ↦ drzwi zamknięte na klucz
- ↦ niepokonane widmo
- ↦ widmo, które może być zabite przez dotknięcie krzyża lub napoju

Użyj klepsydry, aby wziąć napój

Podziemie

Możesz przejść przez ten mur

Zepchnij ciało i zeskocz, nie przejmuj się, że jest wysoko

Przejście do tego miejsca

Księga Światła

BAJTKOWA LISTA PRZEBOJÓW 10

Niektóre z [illegible] wracają na naszą lis[illegible]nnicy nie darowali s[illegible]padk[illegible]tą zobaczymy sami

1	KNIGHT LORE	⇧
2	THE WAY OF THE EXPLODING FIST	⇧
3	ATICK-ATAK	(powrót)
4	HOBBIT	⇩
5	JUMPING JACK	⇩
6	SABRE WULF	(powrót)
7	TAPPER	(powrót)
8	ANT ATTACK	!
9	ALIEN 8	!
10	DUN DARACH	!

Tej dziesiątki wśród listów nie znaleźliśmy. Przyznajemy natomiast nagrodę za najlepsze opisy typowanych gier. Ufundowała ją firma EUROBIT — jest to drążek sterowy Quick Shot II, a wygrała go Natasza Błaszczak z Bydgoszczy. Jednocześnie otrzymuje ona tytuł „Bajtkowej Królowej Gier".

Sławek

Nasz adres:
BAJTEK
ul. Wspólna 61
00-687 Warszawa
LISTA PRZEBOJÓW

FIGHTER PILOT

Digital Software

Nie jest to co prawda symulator lotu amerykańskiego myśliwca „przewagi powietrznej" — F-15 Eagle — co sugeruje tytułowa grafika, ale i tak każdy średnio obeznany z sekretami pilotażu człowiek może przy pomocy programu Fighter Pilot (Pilot Myśliwca) stworzyć sobie namiastkę przeżyć współczesnych lotników.

Tym, którzy trzymali już w rękach prawdziwy drążek sterowy samolotu i mają o tym trochę pojęcia, brakować będzie nożnego steru kierunku, tak potrzebnego na przykład w momencie, kiedy prowadzimy walkę kołową z umykającym oraz manewrującym samolotem przeciwnika. W takiej sytuacji przydałyby się również podskrzydłowe klapy, które zmniejszyłyby promień zakrętu naszej maszyny. niestety twórcy programu założyli, że można ich użyć tylko i wyłącznie podczas startu i lądowania. Brakuje również, skoro pilotujemy nowoczesny myśliwiec, rakiet samonaprowadzających się na cel. Do dyspozycji mamy bowiem jedynie działka lotnicze. Ale może to i lepiej, bo umykającego przeciwnika trudno przechwycić, a jeszcze trudniej zestrzelić. A przecież proste gry, zapewniające szybki i łatwy sukces nie wciągają tak mocno...

Twórcom Fighter Pilota udało się mimo wszystko uzyskać wiele elementów zachowań prawdziwej maszyny odrzutowej. Lotnik musi bowiem bezustannie kontrolować szybkość, wysokość, siłę ciągu silnika, a kiedy zdecyduje się przejść z pozycji ucznia pobierającego od komputera lekcję pilotażu do etapu powietrznego pojedynku, musi nauczyć się posługiwać radarem i komputerem pokładowym. Te właśnie niezbędne w każdym nowoczesnym myśliwcu systemy zapewnią naprowadzanie i wizualne spotkanie z samolotem — celem.



Myśliwiec, który prowadzimy nie wchodzi w korkociąg, natomiast przy ściągniętym drążku i zmniejszonych do minimum obrotach silnika „pływa" — tak jak samoloty o układzie skrzydeł delta.

Niecierpliwym podpowiadamy, że nasz przeciwnik pojawia się na samym początku symulowanej walki w kręgu celownika i wówczas strzela się do niego, jak do kaczki. Ale, czy taki sukces satysfakcjonuje?

(wł)

COSMIC WARTOAD

Ocean



Znacie bajkę o królewnie zaklętej w żabkę? W grze Cosmic Wartoad ty jesteś żabą a królewna piękną dziewczyną. Grozi jej śmiertelne niebezpieczeństwo — powoli zjeżdża w dół zawieszona nad jej ciałem piła łańcuchowa... Musisz ocalić królewnę! W nagrodę otrzymasz jej rękę.

Najpierw jednak musisz pokonać labirynt 64 pomieszczeń. W każdym z nich czekają na ciebie części wyposażenia do demontażu fatalnej piły, ładunki energii do laserowego pistoletu, w który jesteś wyposażony, lub ... twoi najwięksi wrogowie — niewinne pisanki, milutkie kurczaczki wraz z mamą-kwoką, bądź stado pszczółek. Nieodłącznym towarzyszem twej wędrówki jest stary ślimak. Bezstronny obserwator przesuwa się w prawo, gdy bierzesz górę nad swymi wrogami, lub w lewo, gdy pisanki rozbijają się na twojej głowie, pszczółki żądlą cię, a milutkie kurczaczki dziobią po nogach. Im bardziej opadasz z sił, tym szybciej ślimak pokonuje drogę do lewego okna. Kiedy tam dojdzie czeka cię śmierć.

Pojawiająca się po każdym zwycięskim pojedynku mapa umożliwia ci zlokalizowanie miejsca, w którym się znajdujesz i wybór następnego (zawsze sąsiedniego) pomieszczenia.

Jeśli udało ci się zdobyć kluczyki do samochodu możesz nim szybko przejechać przez kilka kolejnych „pól bitewnych". Takie przyspieszenia są nie bez znaczenia — na wykonanie swej misji masz zaledwie półtora godziny. Nie wolno ci jednak gnać na złamanie karku, bo zgubisz buty, a na bosaka ratować księżniczki nie uchodzi.

Bądź zatem czujny i szybki. I pamiętaj — tych samych wrogów w niektórych pomieszczeniach łatwiej pokonać, niż w innych. Droga do czekającej na twą pomoc królewny może zatem być łatwiejsza — lub trudniejsza.

(mc)

CEASAR THE CAT

Microrsoft

Był sobie Kot... miłe, puszyste zwierzątko, które tak jak wszystkie prawdziwe koty uwielbia spać, wygrzewać się na piecu, zjeść dobre co nieco i popić mleczkiem.

Ale nasz kot miał pecha: dali mu na imię Cezar — chciał czy nie chciał musiał więc swoją walecznością godnie reprezentować to imię, musiał toczyć ciągłe boje, walczyć i zwyciężać.

A właśnie teraz okazja po temu jest znakomita: armia myszy, która nawiedziła dom jego państwa musi zostać pokonana! Jest to bardzo trudne zadanie i bez naszej pomocy waleczny kocur Cezar na pewno sobie nie poradzi.

Myszy, jak to myszy, szczególnie upodobały sobie dobrze zaopatrzoną spiżarnię, gdzie siedzą przyczajone za słoikiem z powidłami, za pachnącą kiełbaską, albo za jeszcze innym, nie mniej apetycznym obiektem. Cezar musi skradać się niezwykle cicho i ostrożnie, by nie spłoszyć myszy, po czym, z radosnym miauknięciem skoczyć na nią w odpowiednim momencie.

Bieganie po zastawionych słoikami półkach jest bardzo trudne, a strącenie któregoś z nich może być dla Cezara dość nieprzyjemne (chyba, że w uznaniu dotychczasowych zasług — odpowiednia ilość punktów — właściciel daruje mu ten „wypadek przy pracy"). Tak więc widać, że pole walki nie jest łatwe — gdy Cezar podskoczy za wysoko walnie się łebkiem o sufit, gdy zeskoczy z dużej wysokości stłucze sobie łapki (co prawda koty spadają na cztery łapy, ale i te cztery mogą zaboleć!).



FOT. LEOPOLD DZIKOWSKI

Upolowane myszy Cezar musi wynosić na zewnątrz spiżarni specjalnie do tego przeznaczonymi drzwiami. Każda kolejna dziesiątka upolowanych myszy sygnalizowana jest dzwonieniem budzika, a odpowiednią punktację pokazuje waga stojąca na najniższej półce. Waleczny Cezar musi się bardzo spieszyć, bo im więcej czasu upływa, tym więcej smakołyków znika w żarłocznych pyszczkach wroga. Nie może sobie pozwolić nawet na krótką chwilę odpoczynku, bo inaczej będzie musiał zaczynać wszystko od początku.

Pomóżmy Cezarowi pozbyć się natrętnych myszy, bo resztę życia spędzi tułając się po okolicznych śmietnikach. A szkoda by było, gdyż jest to naprawdę miły kocur.

(ap)

ANIMACJA

Animacja obrazów jest jedną z ciekawszych możliwości oferowanych użytkownikowi przez komputery domowe. Szybkie przełączanie stron pamięci (ang. page flipping) jest jedną z technik dzięki której stosunkowo łatwo jest uzyskać wrażenie ruchu obrazów generowanych przez komputer. Poprzez połączenie tej techniki z umiejętnym wykorzystaniem możliwości produkowania kolorowych obrazów, można tworzyć bardzo ciekawe kompozycje. Technika ta stosowana jest jako jedna z wielu w grach komputerowych, a w szczególności przy konstruowaniu programów demonstrujących możliwości graficzne komputera.

Mikrokomputery ATARI oferują szerokie możliwości łatwego tworzenia ruchomych obrazów, co w połączeniu z bogatą kolorystyką stwarza niepowtarzalną szansę dla programistów — eksperymentatorów.

Poniżej zamieszczamy krótki program w BASIC-u, który pozwoli zobrazować użycie techniki przełączania stron i, mamy nadzieję, zachęci czytelników do dalszego eksperymentowania.

Program ten będzie działał na wszystkich komputerach ATARI wyposażonych w minimum 32 K pamięci operacyjnej (800XL, 130XE).

Po wpisaniu programu do komputera ale PRZED uruchomieniem należy go zapisać na taśmie lub dyskietce, aby w razie popełnienia błędu (który może spowodować zablokowanie komputera) nie trzeba było przepisywać programu od nowa.

Obraz, który widzimy na ekranie, jest tworzony i zapamiętywany w odpowiedniej części pamięci RAM komputera. Stąd jest on bezpośrednio przepisywany na ekran poprzez wyspecjalizowany układ komputera zwany ANTIC.

Technika animacji polega na stworzeniu w pamięci komputera szeregu obrazów przedstawiających kolejne fazy ruchu, które są następnie przepisywane na ekran dając wrażenie ruchu. Widać tu wyraźnie analogię do filmów rysunkowych.

Stroną pamięci nazywany jest fragment pamięci RAM o długości 256 bajtów. Cztery strony tworzą odcinek o długości 1024 bajtów, tj. o 1 kB. Liczba stron pamięci koniecznych do zapisania ekranu zależy od tego, w którym trybie graficznym komputer pracuje.

Program ODBIJACZ korzysta z trybu graficznego 5+16 (tzw. tryb 5 bez okna tekstowego) i wymaga 8 stron pamięci dla każdego ekranu.

Im więcej pamięci wymaga dany tryb graficzny tym więcej stron pamięci jest zajętych przez pamięć ekranową, a tym samym mniej ekranów do przełączenia możemy utworzyć. W komputerach ATARI 800XL mamy do dyspozycji (przy korzystaniu tylko z magnetofonu i uwzględniając, że sam program zajmuje 2K) około 36 K pamięci RAM.

Program po wpisaniu i ZAPISANIU na taśmie uruchamiamy instrukcją RUN. Program rozpocznie rysowanie szeregu nieruchomych obrazów, składających się z kolorowych prostokątów, a będących kolejnymi fazami ruchu. Następnie przejdzie do cyklicznego przełączania ekranów co będziemy odbierać jako ruch wykonywany w dwóch płaszczyznach: pionowej i poziomej.

Szybkość ruchu można regulować klawiszami „<" i ">". Przerwanie realizacji programu następuje klawiszem ESC.

ANALIZA PROGRAMU

Realizacja programu rozpoczyna się od ustawienia zmiennych w linii 2000, które będą dalej wykorzystywane w programie jako wielkości odniesienia.

Linie 1000-1010 przygotowują kolejne ekrany (fazy ruchu).

W linii 100 rezerwowanych jest 8 stron pamięci na kolejny ekran począwszy od szczytu dotychczas dostępnej pamięci RAM (komórka 106).

Linie 200-310 wykonują operacje rysowania prostokątów i odbijacza we właściwych dla danego ekranu położeniach.

W linii 1500 następuje ustawienie kolorów tła, odbijacza i prostokątów oraz włączenie ekranu. Ekran może być wyłączony w linii 100 zmieniając instrukcje POKE 559,34 na POKE 559,0 co przyspieszy proces przygotowania ekranów, aczkolwiek sam proces będzie niewidoczny dla użytkownika.

Linie 20-30 są pętlami animacyjnymi. Zmienne H1 i H11 zawierają informacje o numerze pierwszej strony pierwszego i pierwszej strony ostatniego przygotowanego ekranu. Ponieważ nasza animacja składa się z 11 rysunków po 8 stron pamięci każdy, zmienne te służą jako punkty odniesienia dla znalezienia pierwszych stron pamięci pozostałych ekranów. Poprzez zmienianie w pętli zawartości komórki 561 informujemy komputer, na której stronie pamięci znajduje się początek kolejnego ekranu. Tu więc następuje właściwe PRZEŁĄCZANIE ekranów.

Odwołanie do procedury w linii 40-45 pozwala na sterowanie szybkością zmian (ruchu) ekranów oraz na przerwanie realizacji klawiszem ESC.

Można usunąć instrukcje GOSUB 40 z linii 20 i 30 i wówczas będziemy mogli zaobserwować rzeczywiste tempo przełączania ekranów. W tym przypadku jedyną możliwością przerwania realizacji programu jest naciśnięcie klawisza RESET bowiem standardowe przerwanie realizowane klawiszem BREAK jest w programie zablokowane poprzez instrukcje POKE 566,158 w linii 5. Ten sposób programowego zablokowania klawisza BREAK jest najlepszym z wielu możliwych, gdyż w przeciwieństwie do innych opisanych w literaturze nie stwarza konieczności powtarzania instrukcji blokującej. Normalne działanie klawisza BREAK można przywrócić poprzez instrukcje POKE 566,146 lub naciśnięcie klawisza RESET.

podstawie ANTIC 9.85

Marek Kuliński

```
5 REM DO REGULACJI PREDKOSCI SLUZA KLAWISZE '<' I '>' KLAWISZ
'ESC' PRZERYWA DZIALANIE PROGRAMU
10 POKE 566,158:POKE 106,PEEK(740):GOTO 2000
19 REM PETLA ANIMACYJNA
20 FOR L=0 TO 87 STEP 8:POKE 561,H1-L:GOSUB 40:NEXT L:SOUND
0,25,10,14
25 FOR L=0 TO 71 STEP 8:POKE 561,H11+8+L:GOSUB 40:NEXT
L:SOUND 0,33,10,14
30 POKE 77,0:GOTO 20
39 REM REGULACJA SZYBKOSCI
40 SOUND 0,0,0,0:FOR DELAY=1 TO S:IF PEEK(764)<69 THEN GOSUB
50
45 NEXT DELAY:RETURN
50 IF PEEK(764)=54 AND S<70 THEN S=S+0.3
55 IF PEEK(764)=55 AND S>0 THEN S=S-0.3
60 IF PEEK(764)=28 THEN 9999:REM PRZERWANIE PROGRAMU
65 POKE 764,255:RETURN
99 REM PRZYGOTOWANIE EKRANOW
100 POKE 106,T-TX:GRAPHICS 5+16:POKE 559,34:POKE
712,PEEK(53770):LFT=16:RIT=63:BOT=47:TOP=0:TX=TX+8
199 REM RYSOWANIE PROSTOKATOW
200 FOR L=1 TO 12:FOR K=1 TO 2:COLOR K:PLOT LFT,TOP:DRAWTO
RIT,TOP:DRAWTO RIT,BOT:DRAWTO LFT,BOT:DRAWTO LFT,TOP
205 BOT=BOT-1:TOP=TOP+1:LFT=LFT+X:RIT=RIT-XX:NEXT K:NEXT
L:X=X+0.32:XX=XX-0.32
299 REM RYSOWANIE ODBIJACZA
300 COLOR 3:PLOT 38,V:DRAWTO 41,V:DRAWTO 41,VV:DRAWTO
38,VV:DRAWTO 38,V
305 PLOT 39,V-1:DRAWTO 40,V-1:PLOT 39,VV+1:DRAWTO 40,VV+1
310 V=V+4:VV=VV+4:FOR M=1 TO 100:NEXT M:RETURN
1000 GOSUB 100:COLOR 1:PLOT 39,0:DRAWTO 40,0:H1=PEEK(561)
1005 FOR Q=2 TO 11:GOSUB 100:NEXT Q
1010 COLOR 1:PLOT 39,47:DRAWTO 40,47:H11=PEEK(561)
1499 REM USTAWIENIE KOLOROW I WLACZENIE EKRANU
1500 POKE 712,0:POKE 708,146:POKE 709,68:POKE 710,10:POKE
559,34:GOTO 20

1999 REM INICJACJA ZMIENNYCH
2000 DH=PEEK(561): T=PEEK(106):TX=0:V=1:VV=6:X=-0.6:
XX=2+0.6: S=10:GOTO 1000
9998 REM KONIEC
9999 POKE 561,DH:POKE 106,T:GRAPHICS 0:POKE 566,146
```

RENUMERACJA PROGRAMÓW W JĘZYKU BASIC

Firmowy interpreter języka BASIC mikrokomputerów ATARI nie posiada możliwości renumeracji (przenumerowywania) programów. Może to niekiedy być dużym utrudnieniem i narażać użytkownika na stratę czasu. Zdecydowanie najlepszym rozwiązaniem jest korzystanie z interpretera MASIC XL, który oprócz innych cennych możliwości zawiera także instrukcje RENUM do przenumerowywania programów. Co jednak zrobić, jeżeli nie mamy jeszcze BASIC-a XL a pojawia się konieczność renumeracji?

Przedstawiony program pozwala przenumerować linie programu w języku BASIC. Zmienia on numery linii i odwołania do nich (np. w instrukcjach GOTO, GOSUB, TRAP itp.). Program renumerator należy zapisać instrukcją LIST „C:" w przypadku korzystania z magnetofonu, lub instrukcją LIST „D:REUM.LST" w przypadku korzystania ze stacji dyskietek. Jeżeli zachodzi konieczność użycia renumeratora należy go doładować instrukcją ENTER „C:" dla taśmy lub ENTER „D:RENUM.LST" dla dysku. Program przeznaczony do renumeracji powinien kończyć się powyżej linii o numerze 32675 tj. ostatnia linia programu do przenumerowania musi mieć numer mniejszy niż 32675. Linia 32675 to pierwsza linia renumeratora. Renumerator uruchamia się instrukcją GOTO 32675. Renumerator przenumerowywuje program nie przenumerowywując samego siebie.

Mariusz J. Giergiel

```
32675 REM *** PROGRAM RENUMERATOR ***
32676 CLR :POKE 82,2:PRINT CHR$(125);:
PRINT :PRINT "       RENUMERACJA PROGRA
MU":PRINT :PRINT
32677 DIM A000(10),A001$(4)
32678 A001$="A009":A002=0:REM cyfra 9
w tej lini musi byc napisana w INVERSE
 VIDEO !!!
32679 A003=PEEK(130)+256*PEEK(131)
32680 A004=0
32681 FOR A005=1 TO 4
32682 IF PEEK(A003+A005-1)<>ASC(A001$(
A005,A005)) THEN 32686
32683 NEXT A005
32684 A003=PEEK(134)+256*PEEK(135)+8*A
004
32685 GOTO 32691
32686 A004=A004+1
32687 IF A004>127 THEN POKE 82,0:PRINT
 :PRINT "**BLAD WYKONANUA !":POKE 82,2
:PRINT :END
32688 A003=A003+1
32689 IF PEEK(A003-1)>127 THEN 32681
32690 GOTO 32688
32691 RESTORE 32692
32692 DATA 10,12,23,24,13,4,35,27,18
32693 FOR A005=1 TO 9
32694 READ A006
32695 A000(A005)=A006
32696 NEXT A005
32697 A007=PEEK(136)+256*PEEK(137)
32698 A008=A007:A009=0
32699 A010=-1:A011=0
32700 A012=PEEK(A008)+256*PEEK(A008+1)
32701 IF A012>32674 THEN 32710
32702 IF A010<A012 THEN 32706
32703 POKE 82,0:PRINT :PRINT "**BLAD N
ASTEPSTWA po linii nr ";A010:POKE 82,2
:PRINT
32704 LIST A010-1,A010+10
32705 END
32706 A008=A008+PEEK(A008+2)
32707 A011=A011+1
32708 A010=A012
32709 GOTO 32700
32710 PRINT "Znaleziono ";A011;" lini
do przenumerowania":IF A011=0 THEN END

32711 PRINT :PRINT "Podaj poczatkowy n
umer linii ";:INPUT A013
32712 PRINT "Podaj krok renumeracji
 ";:INPUT A014:PRINT
32713 IF A013+A014*A011<32675 THEN 327
15
32714 PRINT "BLAD W DANYCH WEJSCIOWYCH
-przekroczony dozwolony maksymalny num
er linii":END
32715 A008=A007
32716 A012=PEEK(A008)+256*PEEK(A008+1)
32717 IF A012>32674 THEN 32757
32718 A015=A008+PEEK(A008+2)
32719 A016=A008+4
32720 A017=A008+PEEK(A008+3)
32721 FOR A005=A016 TO A017-1
32722 IF PEEK(A005)<>14 THEN 32750
32723 IF PEEK(A005-1)=A000(3) THEN A01
8=1
32724 IF PEEK(A005-1)=A000(4) THEN A01
8=1
32725 IF PEEK(A005-1)=A000(9) AND A018
=1 THEN 32730
32726 FOR A019=1 TO 8
32727 IF PEEK(A005-1)=A000(A019) THEN
32730
32728 NEXT A019
32729 GOTO 32750
32730 FOR A020=1 TO 6
32731 POKE A003+A020+1,PEEK(A005+A020)
:NEXT A020:PRINT "linia";
32732 FOR A021=0 TO 5-LEN(STR$(A012)):
PRINT " ";:NEXT A021
32733 PRINT A012;" odwolanie do linii
";A009
32734 IF A009<32675 AND A009>-1 AND A0
09=INT(A009) THEN 32736
32735 POKE 82,0:PRINT :PRINT "**POZA Z
AKRESEM [ TRAP ? ]":POKE 82,2:PRINT :G
OTO 32750
32736 A022=A007:A004=0
32737 A010=PEEK(A022)+256*PEEK(A022+1)
32738 IF A010=A009 THEN 32746
32739 IF A010>A009 THEN 32743
32740 A022=A022+PEEK(A022+2)
32741 A004=A004+1
32742 GOTO 32737
32743 POKE 82,0:PRINT :PRINT "**BLAD-n
ie istniejaca linia odwolania":A002=A0
02+1:POKE 82,2:PRINT "";
32744 LIST A012:PRINT
32745 GOTO 32750
32746 A009=A013+A014*A004
32747 FOR A020=1 TO 6
32748 POKE A005+A020,PEEK(A003+A020+1)
32749 NEXT A020
32750 NEXT A005
32751 A016=A017+1
32752 IF A016>A015 THEN 32755
32753 A017=A008+PEEK(A017)
32754 GOTO 32721
32755 A008=A008+PEEK(A008+2)
32756 GOTO 32716
32757 A008=A007
32758 FOR A005=1 TO A011
32759 A023=INT(A013/256):A024=A013-256
*A023
32760 POKE A008,A024
32761 POKE A008+1,A023
32762 A013=A013+A014:A018=0
32763 A008=A008+PEEK(A008+2)
32764 NEXT A005
32765 POKE 82,0:PRINT :PRINT "****** K
ONIEC RENUMERACJI ******":POKE 82,2:PR
INT :PRINT "Suma bledow : ";A002:END
```

WERYFIKACJA PROGRAMÓW NA KASECIE

Użytkownicy mikrokomputerów ATARI mający do dyspozycji magnetofon kasetowy jako pamięc masową niejednokrotnie stają przed problemem, czy zapisany właśnie na tasmie program — efekt wielogodzinnej nieraz pracy, zapisał się bez błędów i da się następnie bez kłopotu wczytać. ATARI BASIC nie ma instrukcji pozwalającej na sprawdzenie poprawności zapisu na tasmie. Przedstawiony tutaj program pozwala w znacznym stopniu wyeliminować tę niedogodność.

Program korzysta z tzw. IOCB systemu operacyjnego (Input Output Control Block) sprawdzając komórkę zawierającą kod błędu transmisji. Jeżeli w czasie weryfikacji pojawi się błąd, podany zostanie jego numer, co pozwala niekiedy na usunięcie jego przyczyny. W tym miejscu jedna uwaga: należy używać dobrej jakości taśm żelazowych (IEC BIAS EQ 120us) najlepiej nowych — nie używanych. Niewłaściwe jest stosowanie taśm innego rodzaju (chrom, metal itp.) gdyż magnetofon ATARI nie jest do nich przystosowany i zyskiwane efekty sa gorsze, niż na zwykłej taśmie.

Program, ktorego poprawnosc zapisu na tasmie ma byc przetestowana, musi zaczynać się od numeru 10 lub większego. Jeżeli tak nie jest, to należy go przenumerować (np. specjalnym programem, lub funkcją RENUM w języku BASIC XL). Program weryfikator doładowuje się z tasmy instrukcją ENTER „C:" i uruchamia instrukcją RUN. Jeżeli weryfikacja wypadła niepomyślnie, a nie chcemy mieć programu zapisanego na tasmie razem z weryfikatorem, należy przed ponownym nagraniem zapisu na tasmie skasować linie od 0 do 10, gdzie umieszczony jest program weryfikujący.

Przedstawiony poniżej program WERYFIKATOR należy zapisać na tasmie instrukcją LIST „C:" [illegible] umożliwi jego doładowywanie instrukcją ENTER „C:" do programu [illegible] w pamięci.

Mariusz J. Giergiel

```
0 REM *** WERYFIKATOR    (M.G.) ***
1 ? CHR$(125):CLR :FOR I=1536 TO 1565:
READ A:POKE I,A:NEXT I:TRAP 7:? "
 *** WERYFIKACJA ***"
2 DIM L$(128):POKE 203,ADR(L$)-(INT(AD
R(L$)/256)*256):POKE 204,INT(ADR(L$)/2
56)
3 CLOSE #1:? CHR$(29);"CLOAD czy ENTER
  (C/E)";:INPUT L$:L=255:IF L$="E" THE
N L=0
4 ? CHR$(29);"Prosze przygotowac tasme
,":? "wcisnac PLAY i RETURN."
5 OPEN #1,4,L,"C:":FOR I=0 TO 100000:G
ET #1,B:X=USR(1536)
6 POKE 203,ADR(L$)-(INT(ADR(L$)/256)*2
56):POKE 204,INT(ADR(L$)/256):NEXT I
7 IF PEEK(195)=136 THEN CLOSE #1:? CHR
$(29);"*** W porzadku ***":END
8 ? CHR$(29);"*** BLAD nr - ";PEEK(195
);" ***":END
9 DATA 104,174,138,2,134,61,160,0,162,
0,185,0,4,129,203,203,230,203,208,2,23
0,204,196,61,240,3,76,12,6,96
```

NUMERYCZNA METODA ZNAJDOWANIA WARTOŚCI PIERWIASTKÓW FUNKCJI

Chcemy rozwiązać problem matematyczny przedstawiony równaniem:

$$f(x) = 0$$

gdzie **f(x)** jest dowolną ciągłą funkcją matematyczną np. **sinx − x/3**. Rozwiązaniem problemu są te punkty funkcji, które leżą na osi odciętych (osi X).

Metody numeryczne pozwalają taki problem rozwiązać dla dowolnej funkcji. Prezentowany program napisany na mikrokomputer ZX Spectrum oparty jest na tzw. metodzie siecznych. Jest to metoda optymalna w zestawieniu z innymi metodami numerycznymi znajdywania miejsc zerowych np. „regułą falsi" czy metodą stycznych. Posiada duże możliwości przy małych wymaganiach, jeśli chodzi o dane wejściowe. Danymi wejściowymi do programu są: postać rozpatrywanej funkcji, interesujący nas przedział zmiennej x (jest on określony punktem początkowym x_p i punktem końcowym x_k), w którym mają być znalezione pierwiastki oraz żądana dokładność obliczeń (EPS).

Program nie tylko oblicza pierwiastki, ale również wykreśla funkcję w rozpatrywanym przedziale, zaznacza na wykresie położenie osi X i Y (liniami przerywanymi) oraz położenie znalezionych miejsc zerowych.

METODA SIECZNYCH

Zależność iteracyjna metody sprowadza się do wzoru:

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n) \quad n = 2,3,4,...$$

Metoda pozwala na rozpoczęcie obliczeń od dwóch dowolnych punktów x_1, x_2, przy czym zaleca się, aby były położone możliwie blisko spodziewanego pierwiastka. Proces zbliżania się poszczególnych iteracji x_n do rzeczywistego pierwiastka funkcji przedstawia rys. 1.

Przez punkty na krzywej f(x) o współrzędnych $[x_1, f(x_1)]$ oraz $[x_2, f(x_2)]$ przeprowadza się pierwszą sieczną (1) do przecięcia z osią x, co wyznacza wartość x_3. Po obliczeniu $f(x_3)$ przez punkty $[x_2, f(x_2)]$ i $[x_3, f(x_3)]$ przeprowadza się drugą sieczną (2) otrzymując punkt x_4 itd. Obliczenia kończy się gdy różnica $x_{n+1} - x_n$ jest mniejsza od zadanej dokładności obliczeń EPS (patrz rys. 1). Metodą siecznych można wyznaczyć pierwiastki wielokrotne, a także pierwiastki położone w punktach przegięcia lub w ich otoczeniu.

OMÓWIENIE PROGRAMU

Pamiętajmy, że wartość rozpatrywanej funkcji w zadanym przedziale musi być skończona.

Po wczytaniu danych wejściowych (na EPS najczęściej wczytuje się wartość w granicach 10^{-7} − 10^{-5}), program dzieli zadany przedział na 160 podprzedziałów i przeszukuje je. Gdy stwierdzi, że w danym podprzedziale znajduje się miejsce zerowe stosuje metodę siecznych i wylicza go. Po czym przechodzi do następnego podprzedziału. Obliczone wartości pierwiastków są wyświetlane na bieżąco na monitorze. Ten odcinek programu kończy się komunikatem „To są wszystkie znalezione pierwiastki".

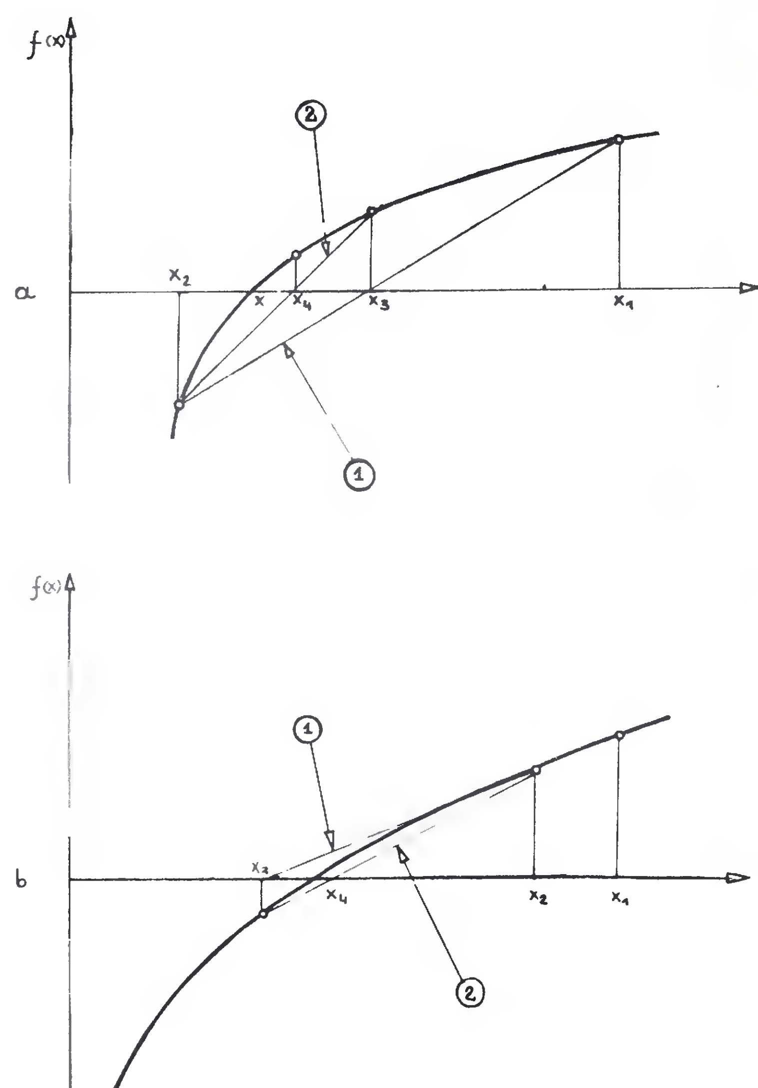
Następny etap programu to przygotowanie danych do wykonania wykresu funkcji. Składa się on z dwóch części:

1) Pierwsza to poszukiwanie wartości minimalnej i maksymalnej rozpatrywanej funkcji w zadanym przedziale
2) Druga to obliczenie współrzędnych punktów funkcji na monitorze.

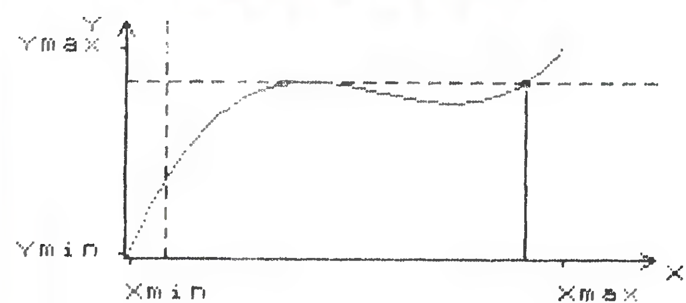
Ten z kolei etap kończy się wyświetleniem na monitorze pięciu opcji wybieranych klawiszami „W", „L", „P", „F" i „2". Naciśnięcie klawisza „W" spowoduje narysowanie osi 0X i 0Y oraz wykresu rozpatrywanej funkcji w zadanym przedziale z zaznaczeniem położeń miejsc zerowych (jeśli zostały znalezione). Po naciśnięciu ENTER program ponownie wypisuje znalezione pierwiastki i ponownie proponuje jedną z wyżej wymienionych opcji.

Przykład przebiegu funkcji sinx − x/3 w przedziale [−6, 6] wraz ze znalezionymi pierwiastkami pokazuje rys. 4.

Inny przykład dla funkcji $x^3 - 9x^2 + 24x - 20$ przedstawia rys. 2. Został znaleziony jeden pierwiastek = 5. Z przebiegu funkcji podejrzewamy, że w lokalnym maksimum jest jeszcze jeden, tym razem podwójny pierwiastek. Aby tę hipotezę sprawdzić naciskamy klawisz „2" i wczytujemy położenia dwóch punktów położonych możliwie blisko spodziewanego wyniku, najlepiej na jednej z gałęzi maksimum. Spełnienie ostatniego warunku wyeliminowuje możliwość przypadkowego wyboru takich dwu punktów,



Rys. 1 Metoda siecznych
a) pierwiastek x znajduje się między dwoma punktami początkowymi x_1 i x_2
b) punkty początkowe znajdują się po jednej stronie pierwiastka.
Widać, że obie iteracje przebiegają różnie, ale obie są zbieżne do wartości pierwiastka.



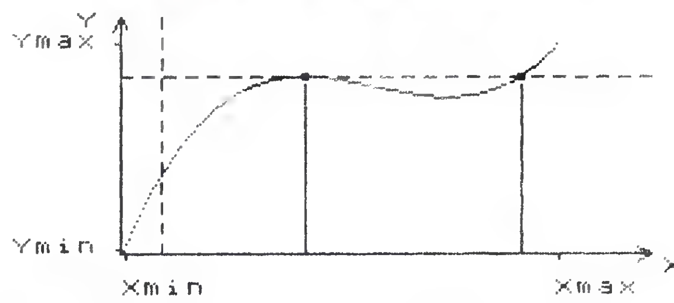

```
Xmin =-0.5         Xmax= 5.51
Ymin =-34.375      Ymax= 6.28325

PIERWIASTKI

x1 = 5

KLAWISZ W - wykres
KLAWISZ L - wylistowanie funkcji
KLAWISZ P - powrot
KLAWISZ F - nowa funkcja
KLAWISZ 2 - sprawdzenie czy jest
 podwojny pierwiastek
```

Rys. 2 — Przykład przebiegu funkcji $x^3 - 9x^2 + 24x - 20$, w przedziale (-0,5, 5.51). Sprawdzenie hipotezy istnienia pierwiastka podwójnego. Pierwszy wykres funkcji



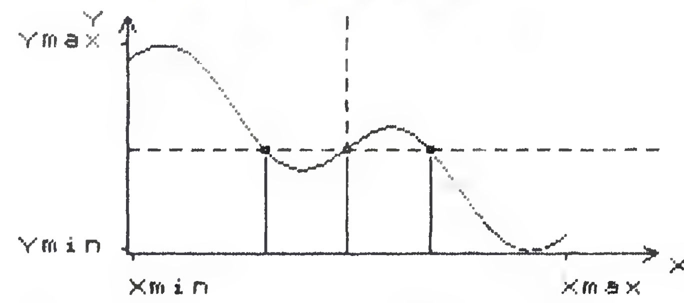

```
Xmin =-0.5         Xmax= 5.51
Ymin =-34.375      Ymax= 6.28325

PIERWIASTKI

x1 = 5
x2 = 1.9999656

KLAWISZ W - wykres
KLAWISZ L - wylistowanie funkcji
KLAWISZ P - powrot
KLAWISZ F - nowa funkcja
KLAWISZ 2 - sprawdzenie czy jest
 podwojny pierwiastek
```

Rys. 3 — Drugi wykres funkcji po tym jak hipoteza okazała się prawdziwa




```
Xmin =-6           Xmax= 6
Ymin =-2.6265338   Ymax= 2.62653

PIERWIASTKI

x1 = 0
x2 = -2.2788627
x3 = 2.2788627

KLAWISZ W - wykres
KLAWISZ L - wylistowanie funkcji
KLAWISZ P - powrot
KLAWISZ F - nowa funkcja
KLAWISZ 2 - sprawdzenie czy jest
 podwojny pierwiastek
```

Rys. 4 — Przykład przebiegu funkcji sinx – x/3 w przedziale (−6,6)

dla których przeprowadzona sieczna byłaby bardzo bliska równoległej do osi X. Gdyby się jednak tak zdarzyło i program zatrzymałby się, należy go ponownie uruchomić przez rozkaz GOTO 520. Wówczas program powróci do punktu wybierania opcji. W wyborze dwu możliwie bliskich spodziewanego rozwiązania punktów pomaga opcja listowania wartości funkcji wybierana przez klawisz „L".

W naszym przykładzie otrzymaliśmy wynik: jest pierwiastek podwójny w punkcie x = 1,999966 (dokładny wynik wynosi x = 2).

Teraz po ponownym narysowaniu wykresu (przez naciśnięcie „W"), mamy już zaznaczone na wykresie wszystkie miejsca zerowe rozpatrywanej funkcji w zadanym przedziale (patrz rys. 3).

Dwie pozostałe nie omówione jeszcze opcje, to:

— Naciśnięcie klawisza „P" — umożliwia ponowne rozpatrzenie tej samej funkcji, ale w innym przedziale
— Naciśnięcie klawisza „F" — uruchamia program od początku, umożliwiając wczytanie nowej funkcji

Program może być również wykorzystywany do badania przebiegu funkcji.

UWAGA PROGRAMIŚCI
Proszę zwrócić uwagę na sposób wykorzystania funkcji VAL w programie.

Tomasz Kostryński

```
  10 REM Miejsca zerowe funkcji
  20 INPUT "Upisz funkcje f(x)=";a$
  30 DIM x(161): DIM y(161): DIM p(161)
  40 DIM a(161): DIM b(161)
  42 LET zf=15-INT (LEN a$/2)
  44 IF zf<0 THEN LET zf=0
  50 CLS : PRINT "Rozwiazujemy rownanie:"'AT 1,zf;a$;"=0"
  60 INPUT "Xp=";xp;"          ";"Xk=";xk;"          ";"EPS=";eps
  70 PRINT "Xp=";xp;"          ";"Xk=";xk;"          ";"EPS=";eps
  80 LET r=1
  90 FOR k=1 TO 161
 100 LET x(k)=xp+(xk-xp)*(k-1)/160
 110 LET x=x(k)
 120 LET y(k)=VAL a$
 130 IF y(k)=0 THEN PRINT "PIERWIASTEK =";x(k): LET p(r)=x(k): LET r=r+1
 140 NEXT k
 160 LET k=1: LET h=0
 160 IF (y(k)<0 AND y(k+1)>0) OR (y(k)>0 AND y(k+1)<0) THEN GO TO 190
 170 IF k=160 THEN GO TO 310
 180 LET k=k+1: GO TO 160
 190 LET c=x(k): LET d=x(k+1)
 200 LET x=d: LET fd=VAL a$
 210 LET x=c: LET fc=VAL a$
 220 LET Rz=d-fd*(d-c)/(fd-fc)
 230 IF ABS (100*(d-Rz))<eps THEN GO TO 270
 240 IF ABS (d-Rz)>1e8 THEN GO TO 260
 250 LET c=d: LET d=Rz: GO TO 200
 260 PRINT "PIERWIASTKA NIE MA w granicach"'c,d: GO TO 290
 270 PRINT "PIERWIASTEK =";Rz: BEEP .5,0
 280 LET p(r)=Rz: LET r=r+1
 290 IF h>0 THEN GO TO 520
 300 GO TO 180
 310 PRINT "   To sa wszystkie znalezione          pierwiastki": BEEP .5,0: BEEP .5,4: BEEP .5,7
 320 DEF FN W(p,o)=(p+o-ABS (p-o))/2
 330 DEF FN M(p,o)=(p+o+ABS (p-o))/2
 340 LET xmin=x(1): LET xmax=x(1)
 350 LET ymin=y(1): LET ymax=y(1)
 360 FOR n=2 TO 161
 370 LET xmax=MEM(xmax,x(n))
 380 LET ymax=MEM(ymax,y(n))
 390 LET xmin=FN W(xmin,x(n))
 400 LET ymin=FN W(ymin,y(n))
 410 NEXT n
 420 PRINT "▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
 430 PRINT "Xmin= ";xmin."Xmax = ";xmax
 440 PRINT "Ymin= ";ymin,"Ymax = ";ymax
 450 PRINT " "'"▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
 460 LET xr=xmax-xmin
 470 LET yr=ymax-ymin
 480 FOR n=1 TO 161
 490 LET a(n)=INT (160*(x(n)-xmin)/xr)+50
 500 LET b(n)=INT (80*(y(n)-ymin)/yr)+50
 510 NEXT n
 520 PRINT " "
 530 PRINT "KLAWISZ "; INVERSE 1;"W"; INVERSE 0;" - wykres"
 540 PRINT "KLAWISZ "; INVERSE 1;"L"; INVERSE 0;" - wylistowanie funkcji"
 550 PRINT "KLAWISZ "; INVERSE 1;"P"; INVERSE 0;" - powrot"
 560 PRINT "KLAWISZ "; INVERSE 1;"F"; INVERSE 0;" - nowa funkcja"
 570 PRINT "KLAWISZ "; INVERSE 1;"2"; INVERSE 0;" - sprawdzenie czy jest podwojny pierwiastek"
 580 BEEP 1,4
 590 PAUSE 0
 600 LET b$=INKEY$
 610 IF b$="w" THEN GO TO 730
 620 IF b$="l" THEN GO TO 670
 630 IF b$="p" THEN GO TO 50
 640 IF b$="f" THEN RUN
 650 IF b$="2" THEN LET h=1: INPUT "Granice : ";c;" - ";d: GO TO 200
 660 GO TO 590
 670 CLS : PRINT "    X","Y"
 680 FOR n=1 TO 161
 690 PRINT x(n),y(n)
 700 NEXT n
 710 GO TO 520
 720 REM WYKRES FUNKCJI
 730 CLS : PRINT AT 0,7; INVERSE 1;" PRZEBIEG FUNKCJI "
 736 PRINT AT 1,zf; INVERSE 1;"y=";a$
 740 PLOT 48,140: DRAW 0,-92: DRAW 196,0: DRAW -5,3
 750 PLOT 244,48: DRAW -5,-3
 760 PLOT 48,140: DRAW 3,-5
 770 PLOT 47,50: PLOT 46,50: PLOT 50,47: PLOT 50,46
 780 PLOT 47,130: PLOT 46,130: PLOT 210,47: PLOT 210,46
 790 PLOT 48,140: DRAW -3,-5
 800 PRINT AT 5,1;"Ymax";AT 15,1;"Ymin";AT 17,6;"Xmin";AT 17,26;"Xmax";AT 4,4;"Y";AT 16,31;"X"
 810 PRINT AT 19,0;"Xmin =";xmin;AT 19,19;"Xmax= ";xmax
 820 PRINT "Ymin =";ymin;AT 20,19;"Ymax= ";ymax
 830 FOR n=1 TO 161
 840 PLOT a(n).b(n): BEEP .01.0
 850 NEXT n
 860 IF r<2 THEN GO TO 1000
 870 LET ya=50+INT (80*(0-ymin)/yr)
 880 FOR n=0 TO 24
 890 PLOT 48+n*8,ya: DRAW 4,0
 900 NEXT n
 910 FOR n=1 TO r-1
 920 LET xa=50+INT (160*(p(n)-xmin)/xr)
 930 PLOT xa,ya+1: PLOT xa,ya-1
 940 PLOT xa-1,ya+1: PLOT xa+1,ya-1
 950 PLOT xa-1,ya-1: PLOT xa+1,ya+1
 960 PLOT xa-1,ya: PLOT xa+1,ya
 970 PLOT xa,ya-3: DRAW 0,51-ya
 980 BEEP 1,24
 990 NEXT n
1000 IF xmin<=0 AND xmax>=0 THEN GO SUB 1090
1010 INPUT "Nacisnij ENTER   ";f$
1020 CLS
1030 IF r<2 THEN PRINT "NIE MA PIERWIASTKOW"'" ": GO TO 520
1040 PRINT "PIERWIASTKI"'" "
1050 FOR n=1 TO r-1
1060 PRINT "X";n;" = ";p(n)
1070 NEXT n
1080 GO TO 520
1090 LET xa=50+INT (160*(0-xmin)/xr)
1100 FOR n=0 TO 11
1110 PLOT xa,48+n*8: DRAW 0,4
1120 NEXT n: RETURN
1130 REM KONIEC
```

READY CZY GOTOWY

Angielskie komunikaty wypisywane przez Commodore 64 na ekranie monitora możemy zamienić na ich polskie odpowiedniki. Czy warto? Odpowiedź zostawiamy czytelnikowi. Jako ciekawostkę podajemy natomiast sposób zastąpienia komunikatu READY (stwierdzającego gotowość do pracy) słowem GOTOWY.

Po włączeniu komputera wpisujemy następującą sekwencję. FOR I=40960 TO 49151:POKE I, PEEK (I):NEXT:POKE 1,54 przepisywania interpretera BASIC'a z pamięci ROM do RAM. Następnie wpisujemy drugą sekwencję:
FOR I=41848 to 41853:POKE I,32:NEXT kasowanie nazwy instrukcji READ. Mozemy teraz wpisać nazwę polską GOTOWY. Funkcja komunikatu GOTOWY jest identyczna jak READY.
A$= "GOTOWY": FOR I=1 TO 6:POKE 41847+I,ASC (MID$ (A$,I,1)): NEXT Teraz pokazywać się będzie nazwa polska GOTOWY

Marek Bednarczyk

GIEŁDA

Komis przy ul. Nowy Świat 26 w Warszawie

Amstrad 464 (monitor monochromatyczny)	300 tys.
Commodore 116	60 tys
Commodore Plus 4	130 tys.
Commodore 16 (z magnetofonem)	100 tys.
Commodore 128	450 tys
Sharp MZ 721 (wbudowany plotter)	200 tys.
Spectrum 48 kB	100 tys.
Spectrum Plus	150 tys.
Magnetofon do Spectrum — Philips	48 tys.
Drążek sterowy (Joystick) f-my Elite	10 tys

Komis przy Pl. Dzierżyńskiego w Warszawie

Spectrum Plus	130 tys.
Spectrum 48 kB	100 tys
Atari 800XL (z magnetofonem)	220 tys
Commodore VIC-20	90 tys
Sharp MZ 721 (wbudowany plotter)	240 tys.

Sklep ABC (stoisko P.P. BOMIS) w Warszawie

Spectrum Plus	210 tys
Commodore 128	680 tys.
Amstrad 6128 (monitor monochromatyczny)	1 mln
Amstrad 6128 (monitor kolorowy)	1.150 tys.
Amstrad PCW 8256	1.890 tys.
Dyskietki 5 1/4	3.2 tys

Salon UNITRY (stoisko f-my UNIPOLBRIT) przy ul. Górczewskiej w Warszawie

Unipolbrit	213.5 tys.
Unipolbrit	280 tys
Monitor kolorowy Neptun	241.5 tys.

Dla porównania podajemy ceny sprzętu i oprogramowania na giełdzie komputerowej w Szkole Podstawowej nr 25 w Warszawie (róg Grzybowskiej i Marchlewskiego):

Amstrad 464 (monitor monochromatyczny)	220 tys
Amstrad 464 (monitor kolorowy)	320 tys.
Atari 800 XL (z magnetofonem)	120 tys
Spectrum 128 k	200 tys.
Spectrum 48 k	80 tys
ZX-81	30 tys.
Drążek sterowy (Joystick)	9 tys
Interface do Spectrum pracujący w obcji Kempstona	9 tys.
Dyskietki 3 " (do Amstrada)	7 tys

Sławomir Polak



PEEK i POKE w C64

Rozkaz PEEK (N) umożliwia odczyt zawartości N-tej komórki pamięci, natomiast rozkaz PRINT PEEK (N) powoduje wyświetlenie zawartości N-tej komórki pamięci na ekranie telewizora (monitora). Liczba N stanowi adres komórki pamięci i jest liczbą dziesiętną z zakresu od 0 do 65535. Zawartość do wolnej komórki pamięci przedstawiona jest na ekranie jako liczba dziesiętna od 0 do 255 (co odpowiada liczbom binarnym od 00000000 do 11111111).

Rozkaz POKE N,X powoduje zapis dowolnej liczby dziesiętnej X (od 0 do 255) w komórce pamięci o adresie N.

Znajomość tych rozkazów umożliwia tworzenie najróżniejszych sztuczek w programie.

Zabawa w kolory

POKE 53280,X — zmienia kolor ramki obrazu
POKE 53281,X — zmienia kolor tablicy obrazu
POKE 53280,PEEK (53281) ujednolica kolor ramki i tablicy
POKE 646,X — zmienia barwę kursora
POKE 646,X,PEEK (53281) kursor ma taką samą barwę jak tablica (jest niewidoczny)
POKE 53265,80 — kursor jest czerwony a zapis pozostaje jasnoniebieski
POKE 56325,X — zmienia częstotliwość migotania kursora
POKE 56325,0 — zwalnia pracę komputera
POKE 56325,51 — normalna szybkość pracy komputera

Napiszmy teraz następujący program:

```
5 POKE 56325,0
10 R=53280:T=53281
20 FOR X=0 TO 255
30 POKE R,X
40 FOR Y=0 TO 255
50 POKE T,Y
60 PRINT CHR$ (147) CHR$ (5)
70 PRINT"NUMER KOLORU RAMKI ";X
80 PRINT"NUMER KOLORU TABLICY ";Y+1
90 NEXT Y,X
```

Sposoby na ciekawskich

Istnieje wiele sposobów zabezpieczenia programu przed wyświetlaniem go na ekranie telewizora (monitora). A oto kilka przykładów

POKE 775,200 — blokuje rozkaz LIST
POKE 775,1 — też blokuje rozkaz LIST
POKE 775,167 — uaktywnia rozkaz LIST
POKE 774,0 — zostają tylko wyświetlone numery linii programu
POKE 774,26 — wszystko znowu normalnie
POKE 774,100 — rozkaz LIST spowoduje SYNTAX-ERROR
POKE 774,27 — jak w kotle groch z kapustą
POKE 22,35 — wyświetla program bez numeru linii
POKE 22,25 — a teraz wszystko razem
POKE 777,1 — komputer nie przyjmuje żadnego rozkazu
POKE 120,2 — też nie słucha rozkazów
POKE 770,100 — a zatrzymaj mnie teraz jak potrafisz
POKE 768,143 — po skończeniu programu powoduje RESET
POKE 53265,11 — obraz znika, lecz jego zawartość pozostaje w pamięci
POKE 53265,27 — obraz się pojawia

Jeśli w programie umieścimy poniższą sekwencję rozkazów

1 POKE 774,226:POKE 775,252

to po rozkazie LIST nastąpi zniszczenie programu. Ciekawym zabezpieczeniem przed wyświetleniem programu jest napisanie linii:

1 REM L

gdzie L to naciśnięcie jednoczesne klawiszy L i SHIFT

Blokowanie i odblokowywanie klawiatury

POKE 649,0 — blokada działania klawiatury
POKE 649,10 — a teraz już klawiatura pracuje normalnie
POKE 655,71 — tak samo jak POKE 649,0
POKE 655,72 — i znów pracuje normalnie
POKE 808,239 — klawisz RUN/STOP nie reaguje
POKE 808,225 — a teraz klawisz RUN/STOP–RESTORE
POKE 808,251 — też blokuje RUN/STOP–RESTORE
POKE 808,237 — a teraz wszystko funkcjonuje normalnie
POKE 788,52 — [illegible] RUN/STOP
POKE 788,49 — odblokowuje RUN/STOP
POKE 792,193 — blokuje RESTORE
POKE 792,71 — odblokowuje RESTORE
POKE 792,226:POKE 793,252 po naciśnięciu klawisza RESTORE [illegible] RESET
POKE 788.226:POKE 789,252 [illegible] naciśnięciu klawisza RE[illegible] RESET

Ochrona przed piratami

Chcąc uniemożliwić zapis programu na kasecie czy dysku należy napisać sekwencję rozkazów

POKE 818,253:POKE 819,253:POKE 808,225

lub

POKE 818.34:POKE 819.253:POKE 808,225

Dla ciekawskich

Chcąc wiedzieć jak długi (ile bajtów) jest program napisany w BASIC-u, znajdujący się w pamięci RAM, należy wpisać następującą sekwencję rozkazów PEEK

PRINT (PEEK (45) +256*PEEK (46)) — (PEEK (43) +256*PEEK (44))

gdzie

PEEK (43) +256*PEEK (44) — adres startowy programu
PEEK (45) +256*PEEK (46) — adres końcowy programu napisanego w BASIC-u a zarazem początkowy adres zmiennych.

Rozkaz POKE 44,PEEK(46)+1 umożliwia obejrzenie zawartości dyskietki LOAD"$",8 bez kasowania istniejącego programu w pamięci RAM. Gdy chcemy znowu powrócić do programu wystukujemy POKE 44,8.

Dla roztargnionych

Rozkaz NEW kasuje program zawarty w pamięci RAM. Nie jest to w pełni prawdziwe. Rozkaz NEW zeruje tylko dwie początkowe komórki pamięci programu o adresie 2049 i 2050 oraz ustawia między innymi we wskaźniku końca programu (adres 45 i 46) wartość 3 i 8 tj. adres początku programu. Możemy po rozkazie NEW znowu odtworzyć postać programu. Dokonujemy tego przez:

POKE 2049,1:POKE 2050,1:SYS42291

Dla muzykalnych

Za pomocą rozkazów POKE możemy uzyskiwać najróżniejsze efekty dźwiękowe. Poniższa tabelka pokazuje wiele wariantów tych efektów. Linie 10 i 20 są zawsze takie same

10 S=54272
20 FOR K=0 TO 24:POKE S+K,0:NEXT

następne wykonujemy wg tabelki.

Marek Bednarczyk

	linia 30	linia 40	linia 50	linia 60	linia 70	linia 80	linia 90	linia 100
	Częstotliwość	Pogłos	Częstotliwość graniczna	Rezonans	Łączenie dźwięku	Rodzaj fali	Pętle	Pętle
Strzał	POKE S+0,0: POKE S+1,18	POKE S+5,1×16+11	POKE S+22,110	POKE S+23,15×16 +3	POKE S+24,5×16 +15	POKE S+4,0: POKE S+4,129	FORJ=1 TO 255: POKE S+0,J: NEXT	FORA=1 TO 1000: NEXT: GOTO 80
Eksplozja	POKE S+0,0: POKE S+1,6	POKE S+5,2×16 +13	POKE S+22,100	POKE S+23,15×16 +3	POKE S+24,3×16 +15	POKE S+4,0: POKE S+4,129	FORJ=1 TO 100: POKE S+0,J: NEXT	FORA=1 TO 4000: NEXT: GOTO 80
Bicie zegara	POKE S+0,0: POKE S+1,6	POKE S+5,1×16 +10	POKE S+22,110	POKE S+23,15×16 +3	POKE S+24,1×16 +15	POKE S+4,0: POKE S+4,17	FORJ=1 TO 255: POKE S+0,7: NEXT	FORA=1 TO 500: NEXT: GOTO 80
Burza	POKE S+0,0: POKE S+1,40	POKE S+5,10×16 +12	POKE S+22,0	POKE S+23,0	POKE S+24,0×16 +15	POKE S+4,0: POKE S+4,129	FORJ=1 TO 255: POKE S+0,J: NEXT	FORA=1 TO 3500 NEXT GOTO 80

WSZYSTKO DLA WSZYSTKICH



PRZEDSIĘBIORSTWO OBROTU MASZYNAMI I SUROWCAMI

Warszawa, ul. Flory 9
oferuje do sprzedaży:

MIKROKOMPUTERY KOMPATYBILNE IBM PC/XT
w cenie około 10 mln zł o konfiguracji:
— jednostka centralna 640 kB z klawiaturą
— dwie stacje dysków 5 1/4 cala
— dysk 20 MB
— monitor monochromatyczny
— drukarka

Zapewniamy serwis i gwarancję na okres 9 miesięcy.

Posiadamy w sprzedaży nie objętej serwisem i gwarancją

MIKROKOMPUTERY:
— Amstrad — Schneider 464, 664, 6128, 8256
— Commodore 64, 128
— Sharp
— ZX Spectrum +
— kompatybilne z IBM

URZĄDZENIA PERYFERYJNE
— drukarki
— Interface do „Spectrum"
— stacje dysków
— dyskietki 3" i 5 1/4"

Informacji udziela:
Zespół Obrotu Surowcami
Warszawa, ul. Flory 9 V p,
tel. 49-01-20, 49-07-91

K-91

amico

AGENCJA KOMPUTEROWA

SOSNOWIEC

PROGRAMY KOMPUTEROWE POCZTĄ, dla ATARI, AMSTRADA, COMMODORA, i SPECTRUM wysyła Agencja Mikrokomputerowa, Sosnowiec P-157, tel. 699-649.

K-76

Wypożyczalnia Programów Komputerowych SDH „Feniks" Warszawa, ul. Żelazna 32 poleca programy użytkowe, edukacyjne, gry, instrukcje wysyłka na cały kraj, katalogi gratis AMSTRAD, ATARI, ZX — SPECTRUM.

D-66

BIURO USŁUG KOMPUTEROWYCH

Pośrednictwo sprzedaży komputerów — programy, części zamienne — Warszawa,

tel. 41-44-48.

D-49

ZELMEVAC service COMPUTECHNIK

Serwis gwarancyjny, informacje i demonstracje komputerów:
— Commodore C-64
— Commodore C-128
— Schneider CPC 464
— Schneider CPC 6128
— Joyce PCW 8256 i PCW 8512
— Urządzenia peryferyjne
— Dyskietki, kasety,
— Gry, programy

01-793 Warszawa ul. Rydygiera 9c
Tel.: 39 05 64 w godz. 9-15 oprócz sobót.

D-73

POLSKIE TOWARZYSTWO CYBERNETYCZNE W POZNANIU ORGANIZUJE ZAOCZNE KURSY: PROJEKTOWANIA SYSTEMÓW INFORMATYCZNYCH (12 mies.), PROGRAMOWANIA EMC W JĘZYKACH FORTRAN, COBOL (10 mies.) i BASIC (8 mies.) ORAZ DLA PROJEKTANTÓW I UŻYTKOWNIKÓW MIKROKOMPUTERÓW (5 mies.). ZAJĘCIA ODBYWAJĄ SIĘ W POZNANIU RAZ W MIESIĄCU PRZEZ 3 DNI.

Informacji udziela i zgłoszenia przyjmuje:
Polskie Towarzystwo Cybernetyczne
Oddział Okręgowy, Ratajczaka 15/15
61-813 POZNAŃ.

K-85

ZX SPECTRUM ATARI

WYPOŻYCZALNIA PROGRAMÓW.
WYSYŁKA NA CAŁY KRAJ.
KATALOGI-GRATIS.
KLUBY — ZNIŻKA.

00-849 Warszawa, UPT 66
skr. p.: 14.

D-54

JAK SIĘ REKLAMOWAĆ W BAJTKU?

Reklamy przyjmuje Młodzieżowa Agencja Wydawnicza (Redakcja Wydawnictw Poradniczych i Reklamy), 04-028 Warszawa, Al. Stanów Zjednoczonych 53, pokój 313. tel. 105682

Cena ogłoszeń: 200 zł za 1 cm plus dodatki za kolor

„MIKROKOMPUTER: ELEMENTY, BUDOWA, DZIAŁANIE"



— jak wskazuje sam tytuł, nie jest to podręcznik programowania, lecz książka, w której wyjaśniono w przystępny sposób podstawowe kwestie związane z mikrokomputerami.

Znajdujemy więc na wstępie, w rozdziale **„Podstawowe pojęcia"**, bliższe omówienie pojęć z dziedziny techniki mikroprocesorowej; dowiadujemy się kolejno, co to jest: arytmometr (jednostka arytmetyczna), układ sterujący, rejestr — by następnie zapoznać się z pojęciem procesora, mikroprocesora i mikrokomputera.

Pamięć, komórki pamięci, rozkazy, dane, argumenty, urządzenia wejścia-wyjścia, lista rozkazów, oprogramowanie, sieć mikrokomputera — oto dalsze pojęcia, bez których znajomości i zrozumienia nie możemy przejść do drugiego, dużego rozdziału jakim jest **„Logika mikrokomputerów"**. Opisano w nim elementarne komórki mikroprocesora, wyjaśniono pojęcie bitu i operacji na bitach, bramki — by następnie przejść do projektowania układów logicznych i układów z pamięcią. Rozdział kończy się opisem przerzutników synchronicznych i związanych z nimi pojęć funkcji wzbudzeń, funkcji wyjść, układów synchronicznych i asynchronicznych.

Rozdział trzeci — to czysta **„Arytmetyka mikrokomputerów"**: na jego podstawie uczymy się dodawania liczb dwójkowych, dowiadujemy się czym jest sumator, długość słowa mikroprocesora, zapis w kodzie uzupełnień do dwu itp. Przesunięcia arytmetyczne i zapis szesnastkowy oraz operacje w kodzie BCD — kończą wspomniany rozdział.

Następne cztery, duże rozdziały poświęcone zostały technice systemów cyfrowych, funkcjonowaniu mikroprocesora oraz organizacji mikrokomputera. Przedstawiona została zasada funkcjonowania mikrokomputera od strony sprzętowej i programowej.

Kilka stron dalej poznajemy kolejne pojęcia z techniki systemów cyfrowych: układy TTL małego stopnia integracji, bloki funkcjonalne, magistrala, układy TTL LS; budowy i działania mikroprocesora: struktura wewnętrzna, rozkazy przesłań i tryby adresowania, rozkazy arytmetyczno-logiczne, skoki, podprogramy, stosy, rozkazy wejścia-wyjścia oraz organizacji mikrokomputera: pamięć i jej budowa, układy wejścia-wyjścia, komunikacja mikroprocesora z pamięcią i układami wejścia-wyjścia, system przerwań mikroprocesora, dołączanie urządzeń zewnętrznych do mikrokomputera i jego programowanie.

W końcowej partii książki znajdujemy schematy blokowe 12 układów scalonych z rodziny UCY 7400, 8 z rodziny UCY 74100, 2 z rodziny LS 240 oraz 3 z rodziny UCY 745400, a także — bardzo ważne! — krótki słowniczek terminów angielskich, od ACCESS do ZERO.

Książka przeznaczona jest dla osób nie związanych zawodowo z techniką komputerową, o przygotowaniu na poziomie szkoły średniej. Mogą z niej więc korzystać wszyscy zainteresowani: uczniowie szkół średnich, studenci młodszych lat (zwłaszcza spoza specjalności zbliżonych do informatyki), także wszystkie te osoby, które z racji własnych zainteresowań lub w związku z pracą zawodową zetknęły się albo mogą się zetknąć z mikrokomputerami.

Andrzej RYDZEWSKI, Krzysztof SACHA „MIKROKOMPUTER: elementy, budowa, działanie" wyd. NOT-SIGMA, Warszawa 1985, wyd. I, nakład 60 370 egz., 160 stron. Cena 250 zł. (jz)

„PROGRAMOWANIE W JĘZYKU BASIC"



Każda literatura — również techniczna — ma swoje bestsellery. Do nich z pewnością należy już drugie (wyd. I ukazało się w 1977 roku), uzupełnione wydanie „PROGRAMOWANIA W JĘZYKU BASIC".

Ta ponad 190 stronicowa, wydana przez Wydawnictwa Naukowo-Techniczne w nakładzie 40 tys. egzemplarzy książka, rozeszła się błyskawicznie wzbogacając biblioteczki miłośników informatyki.

W przedmowie do drugiego wydania autorzy piszą: **„Pomimo nieuchronnego procesu starzenia się, któremu podlega każdy język, można zaryzykować twierdzenie, że BASIC przez ostatnie dwadzieścia lat nie tylko nie stracił na znaczeniu i popularności, lecz wręcz zyskał. Sprawił to rozwój mikroprocesorów, który zaowocował pojawieniem się komputerów osobistych. Prostota BASIC-u, oraz jego wygodny, konwersacyjny tryb pracy powodują, że znakomita większość komputerów osobistych jest wyposażana w pierwszej kolejności w translator tego języka. (...) Mimo, że później zakres zastosowań języka znacznie się rozszerzył, to w dalszym ciągu jest on stosowany do nauki programowania. Obecnie panuje, jak się wydaje, pogląd, że nie jest on w tym względzie narzędziem najlepszym. Nie umożliwia bowiem m.in. tworzenia programów o przejrzystych strukturach oraz ma skromne, w porównaniu do aktualnych języków, typy danych".** Tym należy tłumaczyć m.in. fakt, iż najnowsze modele popularnych komputerów — oprócz powiększonej do 128 kB pamięci — posiadają inny, ulepszony **BASIC** (np. BASIC 7.0 w komputerze Commodore 128) oraz system operacyjny CP/M Plus, pod którym pracuje szereg programów profesjonalnych.

Podręcznik programowania w języku BASIC przeznaczony jest dla szerokiego kręgu zarówno początkujących, jak i doświadczonych użytkowników minikomputerów.

Na początku czytelnik zostaje wprowadzony w tajniki pisania programów i myślenia algorytmicznego, a następnie poznaje bliżej instrukcje języka **BASIC**. Ostatni, krótki rozdział poświęcony jest konkretnej realizacji programów.

Opis języka jest ilustrowany licznymi przykładami i zadaniami do samodzielnego wykonania, których prawidłowe rozwiązanie można łatwo sprawdzić zaglądając do „ściągaczki" na końcu książki.

Można bez przesady powiedzieć, iż z drugim wydaniem, uzupełnionym o dodatki zawierające krótkie opisy dialektów języka **BASIC** dla popularnych w Polsce komputerów osobistych: ZX-Spectrum oraz Meritum I, autorzy trafili w dziesiątkę. Zapotrzebowanie bowiem na tego typu literaturę — a na taki podręcznik w szczególności — jest ogromne.

Zbigniew CZECH, Krzysztof NAŁĘCKI, Stanisław WOŁEK „Programowanie w języku BASIC" wyd. WNT, Warszawa 1985, wyd. II uzupełnione, nakład 40 tys. egz., 190 stron. Cena 170 zł.

Jerzy Zawadzki

„SOWA UCZY LOGO"



Nie, to nie żart. To po prostu kolejna wersja LOGO w języku polskim na mikrokomputer ZX Spectrum. Podręcznik autorstwa Macieja Szewczuka został opracowany w oparciu o przygody Kubusia Puchatka.

Sowa Przemądrzała prowadzi lekcje programowania w języku LOGO dla Krzysia i jego przyjaciół. Wykłada chodząc po polance i zostawiając ślady łapek na piasku a jej koleżanka — Sowa Komputerowa robi to samo na ekranie monitora.

Dlaczego właśnie Sowa? Autor chciał uniknąć dodatkowych trudności stosowania polskich liter (ż, ó, ą itp.) i zamiast słowa „zolw" we wszystkich komendach występuje „sowa". Sowa Komputerowa wykonuje te same czynności co żółw i jest tak samo posłuszna, więc nie ma powodu aby tego rozwiązania nie przyjąć.

Sowa Przemądrzała energiczna, ambitna, z wysokimi aspiracjami troszeczkę przeceniła swoje siły. Zdarzyło się jej kilka drobnych błędów. Mimo to jej wykłady były bardzo interesujące. Zwierzątka szybko zrozumiały zasady zabawy i po skończonych lekcjach zabrały się do samodzielnego programowania.

Zachęcam do przeczytania książki nie tylko dzieci w wieku 6-ciu lat, ale młodzież i dorosłych rozpoczynających zabawę z komputerem.

Do książki dołączona jest kaseta magnetofonowa zawierająca z jednej strony program a z drugiej procedury przydatne do nauki i korespondujące z treścią podręcznika.

Maciej Szewczuk „ZX SPECTRUM, SOWA UCZY LOGO" wyd. TACT ELECTRONICS CORPORATION, Warszawa 1986, 146 stron. Cena wraz z kasetą 3 tys. zł. (Wypożyczalnia programów w Hali Mirowskiej i sklep „COMBO" ul. Słupecka 8 w Warszawie)

Magdalena S[illegible]

JUNIOR I MAZOVIA

Czemu on taki żółty? — pytano złośliwie w komputerowym światku uczestników i gości 58 Międzynarodowych Targów Poznańskich. Do środka Mazovii 1016 zaglądałem tylko pobieżnie, nie jestem więc w stanie definitywnie określić, jaka część jej wnętrza wyprodukowana została na dalekim Tajwanie. Nie jest to zresztą aż tak istotne. Najważniejsze, że „polski IBM", jak ochrzczono już ten komputer, otrzymał na tegorocznych MTP złoty Medal.

Mazovia jest profesjonalnym, 16-bitowym mikrokomputerem będącym de facto odpowiednikiem najbardziej do niedawna rozpowszechnionego na świecie IBM PC/XT. Co prawda teraz, gdy w Polsce rusza jego produkcja, na światowym rynku większą popularnością cieszy się model AT (Advance Technology — Zaawansowana Technologia), jednak opóźnienie polskiego przemysłu komputerowego uległo zmniejszeniu. Producentem Mazovii jest spółka „Mikrokomputery", w której skład wchodzą wiodące zakłady elektroniczne naszego kraju, takie jak Polcolor (Piaseczno) ERA (Warszawa) MERA-BŁONIE, POLON (Warszawa), REFA (Świebodzice), a także Instytut Maszyn Matematycznych, Metronex i UNITRA-PHZ.

Serce Mazovii to 16-bitowy procesor Intel 8086 lub jego radziecki odpowiednik K1810 WM86. Pamięć RAM 256 lub 640 kB, 48 kB, pamięci ROM zawiera BIOS (system operacyjny) i BASIC. Kontroler monitora steruje monitorem monochromatycznym lub kolorowym (według życzeń). Przez złącze równoległe (Centronics) dołączyć można drukarkę D 100 PC MERA-BŁONIE lub ploter. Pamięć na dysku elestycznym 5,25 cala 2 x 360 kB lub 2 x 180 kB. W przyszłości przewidywana jest możliwość dołączenia dysku twardego typu Winchester o pojemności 10 lub 20 MB.

84-klawiszowa klawiatura typu Qwerty. Rozkład znaków na klawiaturze pracującej w trybie polskim jest zgodny z rozkładem standardowej maszyny do pisania, a jej konstrukcja umożliwia wybór jednego z dwóch rozkładów klawiszy np. polski i rosyjski (cyrylica).

Podstawową zaletą Mazovii jest jej kompatybilność (zgodność) z IBM PC/XT. Jest to w chwili obecnej standard światowy, a oprogramowanie dla komputerów IBM-podobnych zajmuje na światowym rynku softwarowym pierwsze (pod względem ilościowym) miejsce. Znaczenie tego faktu nie trzeba chyba nikomu wyjaśniać: całe oprogramowane użytkowe opracowane dla komputerów klasy IBM PC/XT może być bez żadnych przeróbek użytkowane na Mazovii. Aby dodatkowo jej użytkownikom ułatwić życie, przygotowano już spolszczone oprogramowanie. Za jego pomocą można przetwarzać (redagować) na monitorze i przechowywać w pamięci różnego rodzaju dokumenty lub teksty, tworzyć i zarządzać bazami danych, przygotowywać obliczenia ekonomiczne. Według zapewnień producentów w niedalekiej przyszłości Mazovia poprowadzi całą zakładową księgowość, sterować będzie różnego rodzaju eksperymentami i wesprze pracę projektantów (CAD Computer Aided Designe — Komputerowo Wspierane Projektowanie).

Pierwsze moje pytanie skierowane do przedstawicieli spółki „Mikrokomputery" na Targach, dotyczyło wielkości produkcji. Zapewniono mnie, że jeszcze w tym roku wykonanych zostanie 500 egzemplarzy Mazovii. W roku przyszłym produkcja ma zostać zwiększona 10-krotnie. Informacji o docelowej wielkości produkcji nie udało mi się uzyskać. I na zakończenie cena — komplet składający się z jednostki z dwiema stacjami dysków elastycznych, monitora monochromatycznego, klawiatury i drukarki mozaikowej kosztować ma ok. 3 milionów zł.

Gdybyśmy Mazovię 1016 nazwali komputerem dla ojca, to na miano komputera dla syna zasługuje niewątpliwie JUNIOR prezentowany w Poznaniu przez wrocławskie zakłady ELWRO. Powstał on na bazie nagrodzonego w ub.r. na Targach złotym medalem mikrokomputera ELWRO-800. Cicho, bez niepotrzebnego rozgłosu przygotowała go już jesienią ub.r. grupa naukowców z Instytutu Automatyki Politechniki Poznańskiej. Przyświecała im jedna myśl — zrobić dobry komputer dla szkół. Zadanie to wykonali chyba dobrze, skoro Ministerstwo Oświaty i Wychowania zaakceptowało Juniora i postanowiło skierować go do placówek oświatowych jako mikrokomputer edukacyjny.

Co ciekawe, zakłady ELWRO, które produkować będą Juniora, dowiedziały się o jego istnieniu jako ostatnie. Tylko tym chyba wytłumaczyć można fakt, iż prezentowane na Targach modele „zapakowane" były w tymczasową obudowę ... zabawowych organów elektronicznych.

Wiadomością, która ucieszy użytkowników najbardziej rozpowszechnionego w naszym kraju mikrokomputera ZX-Spectrum jest fakt pełnej kompatybilności Juniora z tą maszyną. Kompatybilności — nie znaczy jednak identyczności. Junior jest od Spectrum o wiele, wiele lepszy.

Co prawda ten sam mikroprocesor Z 80A, ale większa, bo 64-kilobajtowa pamięć RAM. Do 24 kB rozbudowano ROM. Junior ma wbudowaną jednostkę sterującą pamięcią na dyskach elastycznych. Co ważniejsze, układy wejścia, wyjścia umożliwiają dołączenie do tego mikrokomputera (bez niezbędnych jak w przypadku Spectrum interfejsów) stacji dysków, drukarki, drążka sterowego (joystica) myszy, pióra świetlnego i innych urządzeń.

Rezydujący w ROM-ie interpreter języka BASIC oraz program obsługi magnetofonu kasetowego są w pełni kompatybilne z programami ZX Spectrum. Także zapis informacji na taśmie magnetycznej (kasecie) oraz adresy urządzeń zewnętrznych są w obu komputerach identyczne. Wszystkie programy napisane dla Spectrum można z magnetofonu bez żadnych przeróbek wczytać i uruchomić na Juniorze. Dostęp do bardzo bogatej biblioteki oprogramowania Spectrum jest zatem nieograniczony!

Standardowo Junior wyposażony ma być w dwie stacje dysków 5,25-calowych. Zarządza nimi dyskowy system operacyjny w pełni kompatybilny z CP/M 2.2.

Juniora wyposażono w 2 wejścia do których można równolegle dołączyć monitor monochromatyczny (zielony) i (lub) telewizor kolorowy pracujący w stosowanym w naszym kraju systemie SECAM. Dotychczas programy ZX Spectrum w pełnej palecie barw mogli oglądać jedynie posiadacze telewizorów wyposażonych w przełącznik PAL/SECAM, gdyż wyjście TV Spectrum przygotowane zostało z myślą o zachodnioeuropejskich posiadaczach telewizorów systemu PAL.

Ważną zaletą Juniorów jest możliwość łączenia ich w sieć mikrokomputerów. Może ona liczyć nawet kilkadziesiąt jednostek. Co daje sieć? Przede wszystkim możliwość wspólnego korzystania z jednej pamięci dyskowej i drukarki przez wielu użytkowników. Pamiętano o nauczycielach, różnicując uprawnienia poszczególnych użytkowników sieci i ustanawiając uprzywilejowaną (nauczycielską) jednostkę sieci. Z takiego uprzywilejowanego Juniora można rozsyłać programy do mikrokomputerowych stanowisk uczniowskich i kontrolować pracę uczniów. Pomyślano również o szkołach, które zakupiły już wcześniej mikrokomputery Sir Clive'a Sinclaira. Równolegle do wyjścia z układu sterowania magnetofonem dołączane może być sprzętowe wyjście umożliwiające dołączenie do Juniora kilkunastu mikrokomputerów Spectrum poprzez ich własne układy komunikacji z magnetofonem.

Dotychczas pisałem wyłącznie o zaletach Juniora i pozytywnych aspektach faktu skierowania go do szkół. Jest jednak również kilka minusów. Podstawowe zastrzeżenia dotyczy wyboru przestarzałego już na światowych rynkach modelu komputera 8-bitowego. Nie jest też chyba najlepszym rozwiązaniem powtórzenie za Spectrum zasady wprowadzenia całych rozkazów za naciśnięciem jednego kluczowego klawisza. Karygodnym błędem jest brak polskich liter.

Zastrzeżenia budzić musi również skala produkcji. Według zapewnień przedstawicieli zakładów ELWRO produkcja Juniora osiągnie w tym roku wielkość 500 egzemplarzy, przy czym wszystkie skierowane mają być do szkół. W przyszłym roku zakłady opuścić ma 5 tys. Juniorów, zaś produkcja docelowa wyniesie rocznie 30 tys. Nie ulega wątpliwości, że w chwili zaspokojenia popytu na te mikrokomputery, będą one anachronizmem takim samym, jakim jest obecnie ZX 81. Jedynie proponowana cena Juniora nie budzi większych zastrzeżeń — 100 tys. zł za sam komputer, choć za pełny zestaw z monitorem monochromatycznym, stacją dysków i drukarką płacić przyjdzie — 800 tys. zł.

I ostatnia już uwaga. Stacja dysków, w które wyposażone będą pierwsze Juniory, pochodzić będą z importu — głównie z NRD i Węgier. Tymczasem w tym samym pawilonie MTP, Krakowska Fabryka MERA-KFAP eksponowała rodzime stacje dysków. W ELWRO twierdzą, że chętnie kupiliby je, ale Kraków nie zapewnia wymaganej wielkości produkcji. Przedstawiciele MERY-KFAP z kolei utrzymują, że wyprodukują jeszcze w tym roku 500 egz. tych stacji — akurat tyle, ile Juniorów opuści mury ELWRO. A dogadać się nie można. Krakowska fabryka produkująca swój własny komputer IMB-podobny KRAK, w specjalizowaniu się w produkcji tylko stacji dysków upatruje zagrożenie dla swych interesów. Wolą więc produkować wszystkiego po trochu. „To asekuracja" — powiedziałem w rozmowie z przedstawicielami MERY-KFAP. „Nie, to tylko ostrożność" — poprawiono mnie.

Obyśmy tylko z nadmiaru ostrożności nie mieli na rynku kilkudziesięciu różnych komputerów rodzimej produkcji, nie kompatybilnych, z unikatowym oprogramowaniem. Teraz gdy na całym świecie karierę robi pojęcie specjalizacji, swoiście pojmowana ostrożność więcej przyniesie nam wszystkim szkody, niż pożytku.

Marek Przybyszewski



SPADOCHRONIARZE



Cześć Maluchy!

Tym razem koniec żartów — zaczynają się schody. Wystarczy już zabawy z Kubusiem Literką czy Liczydełkiem. Dzisiaj napiszemy PRAWDZIWĄ GRĘ KOMPUTEROWĄ. Z prawdziwą grafiką, liczeniem punktów itd.

Z samolotu lecącego nad morzem wyskakują spadochroniarze. Woda jest zimna i kąpiel nie należy do przyjemności a więc dobrze by było, żeby wylądowali na statku. Ty jesteś jego kapitanem. Musisz tak nim sterować, aby jak najwięcej spadochroniarzy znalazło się na pokładzie. A nie jest to łatwe gdyż pojawiają się na niebie w różnych miejscach a w dodatku wiatr zmienia się co chwilę.

Tyle o samej grze. Spójrzmy teraz na to, w jaki sposób jest ona zrealizowana. Z pewnością przyda nam się opis programu, który dokładnie określa co komputer wykonuje w poszczególnych liniach.

Program napisany jest tak, by mógł być uruchomiony na każdym komputerze. Dlatego też wszystkie efekty graficzne uzyskiwane są za pomocą instrukcji **PRINT.**

Jak to jest możliwe? Proszę bardzo, napiszcie taki program:

```
10 CLS
20 FOR i=1 TO 20
30 LOCATE 1,1
40 FOR j=1 TO i
50 PRINT " ";
60 NEXT j
70 PRINT " *";
80 NEXT i
```

Po jego uruchomieniu zobaczymy poruszającą się w prawą stronę gwiazdkę. Prawda, że proste!

Podobny program możemy napisać dla gwiazdki poruszającej się w kierunku pionowym:

```
10 CLS
20 FOR i=1 TO 20
30 LOCATE 1,1
40 FOR j=1 TO i
50 PRINT
60 NEXT j
70 PRINT " "
80 PRINT "*"
90 NEXT i
```

W linijce powyżej gwiazdki umieszczamy spację (miejsce puste) kasującą poprzednio narysowaną gwiazdkę.

W naszym programie ruch spadochroniarzy i statku zrealizowany jest właśnie tak. Statek porusza się tylko w kierunku poziomym, natomiast skoczkowie poruszają się jednocześnie w kierunku poziomym i pionowym. Ruchem statku możemy sterować naciskając klawisze Z — w lewo, X — w prawo (linie 350-390). Oczywiście statek nie może wyjść poza ekran. Spadochroniarze poruszają się w sposób częściowo przypadkowy. W linii 160 losowana jest początkowa pozycja skoczka (pozioma) a następnie w każdym obiegu pętli 170-440 losowana jest zmiana tej pozycji. Niezależnie od pozycji poziomej w każdym obiegu tej pętli skoczek przesuwa się o jedną pozycję w dół.

Oto cała tajemnica tego programu. Został on napisany na Amstradzie 464. Może on być jednak uruchomiony — po niewielkich zmianach — na każdym komputerze. Zmiany dotyczą wymiarów ekranu (linie 120-130), sposobu korzystania z generatora liczb losowych **RND** (linie 160 i 230), odczytywania klawiatury (linia 350) oraz komend: kasującej ekran i umieszczającej kursor w lewym, górnym rogu (linie 100, 180, 4600, 2010). W przypadku Spectrum trzeba dodatkowo zamienić zmienne występujące w pętlach (spad, yspad) na jednoliterowe. Oto poprawiane linie dla najpopularniejszych komputerów:

```
Commodore 64:

100 PRINT ">CLR<";
180 PRINT ">HOME<";
350 GET K$
2010 PRINT ">CLR<";
VIC 20 dodatkowo:
120 LET XEKRANU=21
130 LET YEKRANU=22

ZX Spectrum:

120 LET xekranu=32
130 LET yekranu=21
160 LET xspad=INT(RND*(xekranu-5))
180 PRINT AT 0,0;
230 LET xspad=xspad+0.5-INT(RND*4)/2
zamiana zmiennych:
spad=s
yspad=z

Meritum

90 RANDOM
120 LET XEKRANU=31
130 LET XEKRANU=15
160 LET XSPAD=INT(RND(0)*(XEKRANU-5))
180 PRINT @1,;
230 LET XSPAD=XSPAD+0.5-INT(RND(0)*4)/2
```

OPIS PROGRAMU:

100-140	inicjalizacja, określenie wartości zmiennych
150-470	pętla główna; 10 skoczków
160	losowanie poziomej współrzędnej skoczka
170-440	pętla; jeden skok
180	ustawienie kursora w lewym górnym rogu
190	drukowanie ilości punktów
200-220	pętla; ustawienie współrzędnej pionowej skoczka
230	losowanie kolejnej współrzędnej poziomej skoczka
240-290	rysowanie skoczka
300-320	ustawienie kursora w pozycji współrzędnej pionowej statku (pozycja stała)
330-370	odczytywanie klawiatury, obliczanie poziomej współrzędnej statku
380-400	ustawienie kursora w pozycji poziomej współrzędnej statku
410	rysowanie statku
420-430	sprawdzenie, czy skoczek nie „wyszedł" z ekranu
450	sprawdzenie czy skoczek wylądował
460	kasowanie ekranu
480-510	tablica końcowa i decyzja czy grasz dalej podprogramy:
1000-1030	ustawienie poziomej pozycji skoczka
2000-2050	komunikat o sukcesie

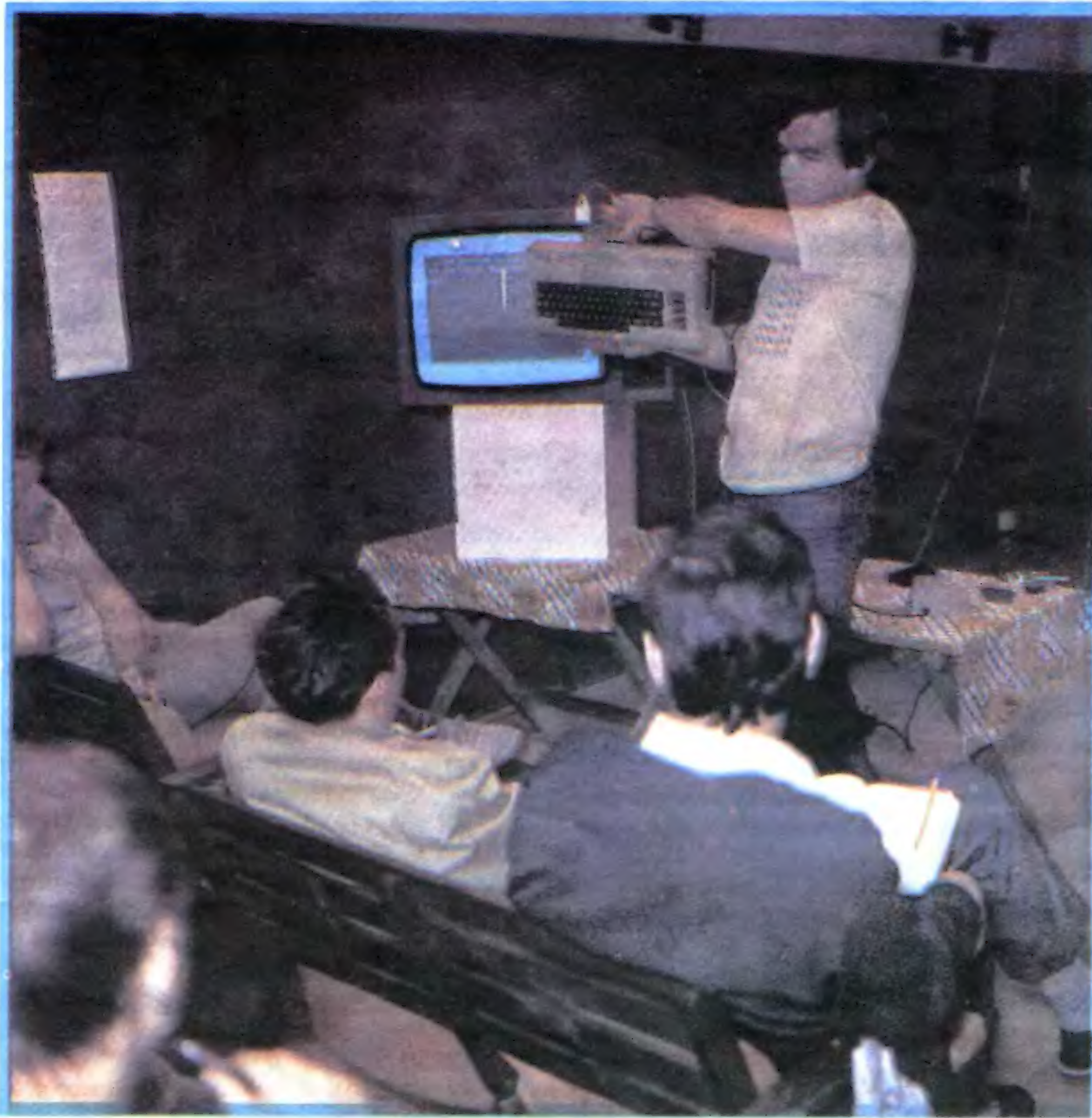
ZMIENNE:

xekranu	— liczba kolumn ekranu — 1
yekranu	— liczba wierszy ekranu — 1
spad	— numer kolejny skoczka
yspad	— współrzędna pozioma skoczka
yspad	— współrzędna pionowa skoczka
yokr	— współrzędna pozioma statku
pkt	— ilość punktów
ks	— znak wczytany z klawiatury
odps	— odpowiedź
t	— czas opóźnienia
x	— współrzędne poziome
y	— współrzędne pionowe

```
99 'inicjalizacja
100 CLS
110 LET pkt=0
120 LET xekranu=39
130 LET yekranu=24
140 LET xokr=xekranu/2-3
149  ' petla glowna - 10 skoczkow
150  FOR spad=1 TO 10
160  LET xspad=INT(RND(1)*(xekranu-5))
169   ' petla - 1 skok
170   FOR yspad=2 TO yekranu-5
180   LOCATE 1,1
190   PRINT "                                  ";p
kt;"pkt."
200    FOR y=1 TO yspad
210    PRINT
220    NEXT y
230   LET xspad=xspad+0.5-INT(RND(1)*4)/
2
240   IF xspad<0 THEN LET xspad=0
250   IF xspad>xekranu-5 THEN LET xspad=
xekranu-5
260   GOSUB 1000
270   PRINT "    "
280   GOSUB 1000
290   PRINT "( )"
300   GOSUB 1000
310   PRINT " , "
320    FOR y=yspad+5 TO yekranu-1
330    PRINT
340    NEXT y
350   LET k$=INKEY$
360   IF k$="z" THEN LET xokr=xokr-1
370   IF k$="x" THEN LET xokr=xokr+1
380   IF xokr<0 THEN LET xokr=0
390   IF xokr>xekranu-7 THEN LET xokr=xe
kranu-7
400    FOR x=1 TO xokr
410    PRINT " ";
420    NEXT x
430   PRINT " \__/ "
440   NEXT yspad
450  IF xspad>=xokr AND xspad<xokr+4 THE
N GOSUB 2000
460  CLS
470  NEXT spad
480 PRINT "Zdobyles ";pkt;"pkt."
490 INPUT "Grasz jeszcze raz (t/n)";odp$
500 IF odp$="t" THEN GOTO 100
510 END
998  '
999  ' podprogramy
1000  FOR x=1 TO INT(xspad)
1010  PRINT " ";
1020  NEXT x
1030 RETURN
1999 '
2000 LET pkt=pkt+1
2010 CLS
2020 PRINT "                  uratowany"
2030  FOR t=1 TO 1000
2040  NEXT t
2050 RETURN
```

Romek

SAMI O SOBIE



MANIAK

15.X.85 w Klubie Osiedlowym na Ursynowie (ul. Wasilkowskiego 7) spotkała się grupa entuzjastów mikroinformatyki — powstał klub Komputerowy „MANIAK".

Jako główny cel klub postawił sobie zadanie upowszechnienia mikroinformatyki oraz bezpłatną wymianę oprogramowania. Wybraliśmy jeden typ komputera — COMMODORE 64, którego popularność jest naszym zdaniem w pełni zasłużona.

Poprzez pojęcie „Upowszechnianie informatyki" rozumiemy przede wszystkim formy działalności zapoznające młodzież (i nie tylko!) z zasadami działania i budową komputerów, nauką języków programowania itp. I na takie właśnie formy działania kładziemy główny nacisk, znane nam są bowiem „kluby" w których „upowszechnia" się mikroinformatykę za pomocą gier komputerowych. Oczywiście organizujemy również i takie imprezy, stanowią one jednak margines naszej działalności.

W ramach działalności dydaktycznej klub organizuje dwukrotnie w ciągu roku 5-miesięczne kursy programowania w języku BASIC dla COMMODORE 64. Oprócz nauki samego języka przedstawiamy wszelkiego rodzaju tricki i usprawnienia dotyczące COMMODORE 64. Dla bardziej zaawansowanych klub prowadzi również zajęcia obejmujące podstawy programowania w języku maszynowym, język „SIMONS BASIC" mapę pamięci C64 itp. Za udział w zajęciach klub pobiera opłatę 250 zł miesięcznie.

Początkowo klub funkcjonował w oparciu o sprzęt prywatny. Od stycznia 1986 r. dzięki naszemu sponsorowi — spółdzielni budownictwa mieszkaniowego „Ursynów" otrzymaliśmy własny sprzęt.

Członkiem klubu „Maniak" może zostać każdy kto wpłaci do klubowego banku programów określoną ich liczbę oraz zobowiązuje się do niehandlowania nimi w jakiejkolwiek formie. W zamian za to można korzystać z całego klubowego banku programów (liczącego obecnie ok. 1000 programów na dyskach i kasetach), zorganizowanego w formie kaset do wypożyczenia: programy dyskowe wymieniamy na miejscu. Spośród ciekawszych pozycji warto wymienić „Assembler Tutor", „Macroassembler", „Printer Snop", „Music Prolessor" itp.

Nie przepadamy za językiem programowania „LOGO". Nie znosimy handlarzy i pseudopośredników z ogłoszeń, oraz przechodzą nas ciarki na myśl o polskiej wersji języka BASIC („wrwsz..."). Z tego też powodu na zajęciach używana jest tylko terminologia angielska. Nie stawiamy żadnych wymogów wiekowych, bowiem czasami siedmio- czy ośmiolatek „chwyta" szybciej aniżeli osoba dorosła.

Efektem 7-miesięcznej działalności klubu są dwa poważne programy klubowe „Maniak Turbo" oraz „BEEP 64". Ponadto powstały również pewne rozwiązania hardware'owe np. sygnalizatory, Autoreset, Kopiarka Taśma-Taśma itp.

W ramach działalności klubowej organizowane są również sesje wymiany oprogramowania, drobnych napraw (np. strojenie głowic czy czyszczenie zestawów klawiatury itp.)

Zajęcia prowadzone są we wtorki i czwartki od 18.00 do 20.00 i w tych dniach można nas zastać w klubie przy ul. Wasilkowskiego 7 (dojazd 504, 483, 486, 134, 136, 177, 185) lub pod telefonem 40-62-64 (prosić Klaudiusza lub Michała). Zapraszamy!

Klaudiusz Dybowski
Michał Silski

SPRZĘŻENIE ZWROTNE



Drogi

...rozumiem, że Spectrum jest najpopularniejszym komputerem, ale nie JEDYNYM. Jeżeli już drukujecie program na Spectrum, to nie zbywajcie użytkowników innych maszyn słowami: „Po adaptacji program można też uruchomić na innym mikrokomputerze". Obok takiego programu powinno znajdować się coś w rodzaju erraty, w której wypisane byłyby linie programów, którymi należy zastąpić istniejące linie w programie na Spectrum, aby program działał na innym, określonym komputerze. Pomijam programy takie, jak np. „Klakson" („Bajtek" nr 1/86), który to program napisany jest tylko i wyłącznie na Spectrum.

Mariusz Szumielewicz
Szczecin

Pomysł bardzo dobry, tylko że... nierealny. Autorzy publikowanych w „Bajtku" programów nie są wszechwiedzący (czego zresztą nie wymagamy od nikogo). Nie możemy od nich żądać, aby po pierwsze: wykazywali się znajomością BASIC'a dla Atari, Spectrum, C64 i CPC464 (to są obecnie najpopularniejsze dialekty), a po drugie: by uruchamiali swój program na wszystkich tych typach komputerów, a tego właśnie wymagałoby sporządzenie sugerowanej „erraty". Ponadto przeróbka programu na inny komputer z reguły pociąga za sobą usunięcie efektów dźwiękowych i graficznych — a zatem gra po prostu nie warta jest świeczki.

Ze swej strony możemy jedynie starać się o zamieszczenie programów według zasady „dla każdego coś innego" co, ze zmiennym być może szczęściem, robimy.

Proszę o informację, gdzie teraz mógłbym kupić 5 pierwszych numerów „Bajtka", na których mi bardzo zależy.

Sławomir Babis
ul. Jeżowska 24
96-200 Rawa Mazowiecka

Listów podobnej treści nadal otrzymujemy sporo, a tymczasem odpowiedź może być tylko jedna: w antykwariacie lub na „Perskim Jarmarku". W redakcji pozostało po kilka egzemplarzy zeszłorocznych numerów, które przechowywane są w... szafie pancernej. Przykro mi, jeżeli rozczarowałem czytelników tą informacją.

Czy programy przeznaczone na Spectrum 48k mogą być używane na Spectrum 16k? Słyszałem, że można program dzielić na bloki.

Rafał Zdon
ul. Z. Jaroszewicz 26/3
58-400 Świdnica

Programy dla Spectrum 48k mogą być używane na Spectrum 16k, o ile wymagają nie więcej niż 16kB pamięci, a jak łatwo zauważyć, w praktyce oznacza to „nie". Owszem, programy można dzielić na bloki, ale wymaga to szybko działającego urządzenia pamięci zewnętrznej — raczej lepszego niż Microdrive. Poza tym tego typu rozwiązanie z góry dyskwalifikuje wszelkiego rodzaju gry polegające na animacji. Dalej, programy w języku wewnętrznym dla Spectrum 48 mogą zawierać skoki bezwzględne dla komórek pamięci o adresach nie istniejących w Spectrum 16k; trzeba by zatem, oprócz wyróżnienia i rozdzielenia bloków programu, dodatkowo zająć się ich lokalizacją w pamięci. Nie polecam nikomu tego rodzaju pracy — zdecydowanie łatwiej poświęcić ten czas na zarobienie pieniędzy w celu rozszerzenia pamięci komputera.

Myślę, że przy obecnej technice drukarskiej złym rozwiązaniem jest publikowanie programów jako kopii wydruków z drukarki. Po prostu drobne literki, jeszcze drobniejsze kropki, przecinki itp. przy złym rozłożeniu farby drukarskiej powodują liczne kłopoty związane z nieczytelnością tekstu: program nie chce „wejść", bo zamiast kropki odczytałem przecinek. To jako przykład. Myślę więc, że lepiej byłoby program drukować tak, jak normalny tekst. Być może zajmie to więcej miejsca, lecz poprawi czytelność.

(...) Z chęcią widziałbym w „Bajtku" wyjaśnienie kwestii: co zrobić, aby po wgraniu własnego programu z taśmy do pamięci, nie trzeba było za każdym razem wciskać RUN + ENTER (w Spectrum), lecz by program uruchamiał się automatycznie?

Ryszard Chustecki
ul. Śląska 10 m. 5
25-328 Kielce

Z dotychczasowej praktyki wynika, że publikowanie tekstów programów w formie fotokopii wydruku sporządzonego na drukarce jest jedynym rozsądnym rozwiązaniem. Z treści Pana listu wnoszę, że zastrzeżenia dotyczyły przede wszystkim starego „Bajtka". Trudno się nie zgodzić, że jakość zeszłorocznych numerów naszego czasopisma pozostawiała wiele do życzenia.

Używanie techniki składu drukarskiego do publikowania programów komputerowych nie jest dobrym rozwiązaniem. Maszynistki i zecerzy nie są bowiem informatykami, dostrzegającymi różnice przy zamianie średnika na dwukropek, wstawieniu lub usunięciu spacji, itp. Człowiek jest istotą omylną, natomiast publikacja kopii wydruku działającego programu daje nam gwarancję jego merytorycznej poprawności. A o to właśnie chodzi!

Aby program napisany w języku SINCLAIR BASIC uruchamiał się automatycznie po wczytaniu go z taśmy, należy go uprzednio nagrać nie za pomocą instrukcji

SAVE „NAZWA"

lecz

SAVE „NAZWA" LINE 10,

o ile pierwsza linia programu nosi numer 10.

Od paru miesięcy jestem posiadaczem mikrokomputera ZX Spectrum 48k, który nabyłem po wielu wyrzeczeniach i trudnościach. Oprócz telewizji, radia i prasy jest on moim jedynym oknem na świat, gdyż jestem inwalidą nie mogącym chodzić. W chwilach wolnych, a mam ich bardzo dużo, wypełniam sobie czas zabawą z komputerem, który, jak na moje dwadzieścia lat, daje mi bardzo dużo zadowolenia i pozwala zapomnieć o kalectwie.

Dużo piszecie na temat ciekawych programów komputerowych. Ja mam tylko trzy gry, które są bardzo proste i nie wymagają większego wysiłku, tak że od pewnego czasu nie

Bajtku!

korzystam z nich. Chciałbym za Waszym pośrednictwem prosić o kontakty z osobami, które są w stanie mi pomóc w wymianie czy też skopiowaniu ciekawych programów.
Nazwisko i adres proszę pozostawić do wiadomości redakcji oraz dla osób, które zechcą mi pomóc w nawiązaniu kontaktów.

Marek L.

Wiele się swego czasu mówiło o wykorzystaniu komputerów do pomocy ludziom niepełnosprawnym — chocby specjalnych urządzeniach pozwalających na porozumiewanie się z niewidomymi i głuchymi, o nauce mowy osób głuchych.

Tymczasem zaś jakby umknęła uwadze najprostsza rola komputera w życiu człowieka niepełnosprawnego — wypełnienie czasu, rozwój zainteresowań, wzbogacenie życia, „okno na świat". Być może my, zdrowi, nie potrafimy tego nigdy zrozumieć.

Tych, którzy chcą pomóc Markowi prosimy o listy.

Marcin

Redakcja Bajtka zebrała już i usystematyzowała wyniki ankiety „Klubowy Bank Danych" z nru 1/86.

Serdecznie dziękujemy wszystkim klubom, które zechciały nawiązać z nami kontakt. Jednocześnie informujemy, że już wkrótce uczestnikom ankiety roześlemy pełny wykaz klubów, o których otrzymaliśmy informacje. Mamy nadzieję, że przyczyni się to do rozwoju kontaktów między nimi.

Wśród nadesłanych ankiet znalazła się również taka:

Nazwa klubu: MALMÖ DATA CLUB

Adres klubu: Münkhattegatan T86. ≠ 91
MALNÖ S–214 74 **SWEDEN**

Agencje w innych miastach: GÖTEBORG — LULEA

Data utworzenia klubu: 1970.10.11

Ilość członków: 320

Posiadane komputery:

ABC–80	10 szt.
IBM 234 AD .	1 szt.
Spectrum 48k	20 szt.
Spectrum 128k	10 szt.
Amstrad PCW 8256	5 szt.
Commodore 64	15 szt.

Czego spodziewacie się po kontakcie z innymi klubami? Wymiany oprogramowania, komponentów, mikroprocesorów, itp. Data: 21 kwietnia 1986 MALMÖ, Szwecja.

NASTĘPCY

Dokończenie ze str. 32

produkowanego modelu spowodowała zmianę koncepcji technologii produkcji i pojawienie się automatów zwanych robotami.

Pierwszego robota z prawdziwego zdarzenia zbudowali w latach 1959—1960 Amerykanie pod wodzą Joe Engelberga, założyciela firmy „Unimation", która przez długie lata dominowała wśród producentów robotów. Roboty tej firmy po raz pierwszy zostały zastosowane w 1961 r. do zgrzewania karoserii samochodowych w fabryce General Motors (nota bene obecnie też jeden z poważniejszych producentów robotów). Pierwszy robot. po przepracowaniu 15 lat, przeszedł już na emeryturę do lokalnego muzeum. Ale prawdziwy rozwój robotyzacji rozpoczął się w latach 1979—82. W tym miejscu trzeba przypomnieć, że pierwszy mikroprocesor 8-bitowy powstał w 1972 r. a 16-bitowy w 1974. Przypominam te fakty dlatego. bo przede wszystkim rozwój mikroelektroniki, a co za tym idzie rosnąca pewność poprawnej pracy robotów w połączeniu ze wzrostem płac robotników, a w niektórych krajach (np. Japonia) wyraźnie odczuwalny niedostatek siły roboczej, stały się przyczyną rozkwitu nowej dziedziny robotyki.

W Polsce prace nad robotami rozpoczęto w połowie lat siedemdziesiątych. Obecnie mamy cztery typy robotów złożonych i kilka typów robotów prostych. Roboty złożone powstały w trzech ośrodkach: w Instytucie Mechaniki Precyzyjnej w Warszawie — RIMP 1000 i RIMP 900, w Przemysłowym Instytucie Automatyki i Pomiarów „MERA-PIAP" — licencyjne roboty IRb i IRb 60, oraz w Centrum Badawczo-Konstrukcyjnym Obrabiarek. w Pruszkowie — PRO 30. Robot PRO 30 po raz pierwszy został wystawiony w 1977 na Targach Poznańskich. Jest on w zasadzie robotem specjalizowanym, przeznaczonym do załadunku i rozładunku obrabiarek. Drugim chronologicznie polskim robotem jest RIMP 1000, którego model powstał pod koniec 1977 r. Jest to robot uniwersalny. Robot do prac malarskich RIMP 900 został skonstruowany w IMP. Powstał stosunkowo niedawno, obecnie trwają badania na stanowisku produkcyjnym. Główną cechą funkcjonalną tego robota jest umiejętność zapamiętywania trajektorii — ruchów wykonywanych przez programującego. Największe nadzieje na zmniejszenie dystansu do światowej czołówki, nie bezpodstawne zresztą, związane były z robotem IRb 6 (60), budowanym na licencji szwedzkiej firmy ASEA. Robot ten, w momencie zakupu licencji (XII 1976 r.) przeszedł długotrwałe próby pracy w przemyśle, podczas gdy pozostałe polskie roboty były na etapie raczkowania. Gdyby licencja ta została wdrożona to dzisiaj Polska być może przodowałaby wśród państw RWPG w dziedzinie robotyki. Niestety, licencja wygasła, a produkcja robotów nie została rozpoczęta.

Choć zewnętrznego podobieństwa między współczesnymi robotami a człowiekiem doszukać się trudno, u obu znajdujemy funkcjonalnie podobne zespoły:
— nośne — szkielet kostny u człowieka; mechaniczny — w robocie,
— napędowe — mięśnie; silniki różnych typów,
— receptory — narządy zmysłów; czujniki,
— sterujące — mózg; komputer.

I właśnie robot jest najwspanialszym zastosowaniem komputera. Jest to wyposażenie elektronicznego mózgu w elementy czuciowe i wykonawcze. Komputer, a raczej układ sterowania robota ma rozwiązywać trzy problemy: ułatwienie człowiekowi planowania zadań dla robota, zapamiętanie wszystkich zaplanowanych operacji i ich automatyczne wykonanie. Łatwo dostrzec. że osnową dla układu sterowania jest program działań robota. Człowiek-operator, w mniejszym lub większym stopniu wspomagany przez komputer, tworzy ten program, „uczy" robota, przekazując mu informacje o tym, w jakiej kolejności i jakie czynności ma wykonać. Najczęściej operator w czasie uczenia steruje robotem ręcznie za pomocą specjalizowanego sterownika lub bezpośrednio, ujmując dłońmi rękojeść przytwierdzoną do ramienia. Układ sterowania, na rozkaz operatora zapamiętuje wybrane położenie ramienia (punkt po punkcie) lub całą przebytą drogę. Nauczanie takie jest dość żmudne i zajmuje dużo czasu. Dlatego coraz powszechniej przy formułowaniu zadań dla robota korzysta się ze specjalnych języków programowania.

Jeszcze bardziej zaawansowana technika programowania polega na opisaniu zadania, jakie robot ma wykonać. Ideałem byłoby, aby można było to zadanie sformułować w potocznym języku np. włóż ten kołek — tu następuje pokazanie kołka — w ten otwór. Zadaniem układu sterowania jest ułożenie planu działania uwzględniającego odszukanie kołka, rozpoznanie go, przeniesienie w pobliże otworu i włożenie. Do tego typu programowania zaprzęga się systemy komputerowe, które pozwalają na ułożenie programu pracy robota i zasymulowanie go bez robota, czyli bez tracenia czasu na miejscu pracy, bez wstrzymywania procesu produkcyjnego. Po ułożeniu programu pracy układ sterowania musi pokierować zespołami napędowymi tak, by robot wykonał zadaną czynność z założonymi parametrami. Wiele nowych możliwości robota realizowanych jest przez naukowców sztucznej inteligencji i nie znajduje na razie ekonomicznego uzasadnienia. Zadaniem robotyki jest rozwiązywanie teoretycznych i praktycznych problemów organizacji kompleksowego zachowania się mobilnego robota wyposażonego w czujniki i elementy wykonawcze. Takim robotom stawia się określone, globalne zadania. Zakłada się jednak, że nie można wcześniej określić ich realizacji, stanu otaczającego go środowiska i dlatego takie problemy, jak planowanie kolejności wykonania zadań, robot musi wykonać samodzielnie.

Rzeczywistą rewolucją przemysłową stał się nie sam robot, ale wprowadzenie elastycznych systemów produkcyjnych. Wyobraźmy sobie fabrykę. w której zainstalowanych jest wiele różnych, wielofunkcyjnych stanowisk dla różnorodnych operacji technologicznych. Stanowiska te są połączone transporterami. Każde stanowisko posiada autonomiczny układ sterowania, który łączy się z innymi siecią komputerową. Do sieci dołączone są sterowniki zarządzające magazynami, komputery planujące i koordynujące. Przy większości stanowisk znajdują się stacjonarne roboty. Uzupełnieniem tego systemu są wózki samojezdne, które rozwożą pojemniki od jednego do drugiego stanowiska. Wózki te mają własny układ sterowania, potrafią poruszać się po ustalonych trasach i nawet przestrzegają zasad ruchu drogowego. Są na tyle niezależne, że gdy stwierdzą rozładowanie akumulatora zasilającego, zjeżdżają do akumulatorowni, gdzie dyżurny robot wymienia akumulator na naładowany. Nie jest to wytwór mojej wyobraźni. Takie fabryki już istnieją, na świecie jest ich około 250, z czego prawie połowa w Japonii. A potęga takiego zakładu tkwi w fakcie, że w chwili zmiany wyrobu, ba — nawet profilu produkcji, nie musimy nic przestawiać, budować nowych linii, kupować nowych maszyn czy zatrzymywać produkcji na kilka miesięcy. Zmianom ulegają programy stanowisk pracy i obsługujących je robotów. Zmieniona zostaje trasa wędrówki wyrobu pomiędzy stanowiskami. Całe przestrajanie fabryki odbywa się przy konsolach komputerów, w pomieszczeniach technologów i programistów.

Kiedy przejrzy się pracę. plany i zrozumie ambicje naukowców, musi nasunąć się nieśmiałe podejrzenie, że następnym gatunkiem w ewolucyjnym rozwoju życia na Ziemi będą... maszyny. Oczywiście, pragniemy, aby człowiek był tworem doskonałym, ostatecznym i niezastąpionym — tak jak wczoraj chcieliśmy, aby Ziemia była centrum wszechświata — niemniej jednak dominujące cechy człowieka powoli będą się wyczerpywały. Dlatego można przyjąć, może trochę szokujące założenie, że jednym z celów człowieka jest stworzenie podstawy, przygotowanie i wprowadzenie nieorganicznej formy bytu.

Andrzej Gogolewski

NASTĘPCY

Maszyny uzupełniające lub wzmacniające ludzkie możliwości stają się maszynami „pomocnikami". Podobnie jak cały świat przeszły one proces ewolucji. Początkowo wykonywały pojedyńcze operacje np. cięcia, toczenie, a rola człowieka sprowadzała się do mocowania detali, uruchomienia odpowiednio nastawionej maszyny itd.

Wraz ze wzrostem skali produkcji maszyny wykonujące pojedyńcze czynności nauczyły się wykonywać cały ciąg operacji.

Jednakże dalszy rozwój automatyzacji natrafił na pewnym etapie na bariery. Pierwsza to bariera techniczna, związana z niezawodnością elementów linii. Druga bariera ma charakter ekonomiczny. Ogromny koszt inwestycji automatycznych linii wymusza długotrwałą ich eksploatację, aby mogły one opłacić się producentowi.

Pierwszą barierę zdołano przełamać przez zastosowanie magazynów międzyoperacyjnych uniezależniających — w pewnym stopniu — działanie linii od krótkotrwałych uszkodzeń jej elementów. Bariery ekonomicznej początkowo nie udawało się przekroczyć. Tempo postępu technicznego wymagało od producenta coraz częstszego wprowadzania nowych odmian i nowych modeli produkowanego wyrobu, co jest sprzeczne z wymaganą długotrwałością eksploatacji automatycznych linii produkcyjnych przystosowanych do produkcji jednego określonego modelu. Przy zmianie modelu należało zmienić całą linię, co było ogromnie kosztowne i wymagało długiego czasu. Zjawiska te można wyraźnie obserwować na przykładzie największych producentów samochodów. Konieczność częstych zmian

Dokończenie na str. 31

Czy następnym gatunkiem w ewolucyjnym rozwoju życia na ziemi będą maszyny?